

81282
p-76

DISTRIBUTED DIGITAL SIGNAL PROCESSORS FOR
MULTI-BODY FLEXIBLE STRUCTURES

FINAL REPORT

to

NASA Langley Research Center
Hampton, Virginia 23665-5225

Project Monitor: Dr. Jer-Nan Juang, M.S. 297

Contract No.

~~NAGW-1-1136~~

NCC1-1136 (SEE LETTER)

Submitted by:

Gordon K. F. Lee
Mars Mission Research Center
North Carolina State University
Raleigh, NC 27695-7910

Date: April 6, 1992

(NAGW-1-1136) DISTRIBUTED DIGITAL SIGNAL
PROCESSORS FOR MULTI-BODY FLEXIBLE
STRUCTURES Final Report (North Carolina
State Univ.) 76 p

CSCL 131

N92-22207

Unclass

63/37 0001282

Executive Summary

Multi-body flexible structures, such as those currently under investigation in the spacecraft design, are large scale (high-order) dimensional systems. As such, controlling and filtering such structures is a computationally complex problem. This issue is particularly important when many sensors and actuators are located along the structures that need to be processed in real-time.

This report summarizes research activity focused on solving the signal processing (that is, information processing) issues of multi-body structures. A distributed architecture is developed in which single loop processors are employed for local filtering and control. By implementing such a philosophy with an embedded controller configuration, a supervising controller (as simple as a workstation or personal computer) may be used to process global data and make global decisions as the local devices are processing local information.

A hardware testbed - a position controller system for a servo motor- is employed to illustrate the capabilities of the embedded controller structure. Several filtering and control structures which can be modelled as rational functions can be implemented on the system developed in this research effort. Thus the results of this study provides a support tool for many Control/Structures Interaction (CSI) NASA testbeds such as the Evolutionary model and nine-bay truss structure.

TABLE OF CONTENTS

	Page
Executive Summary	i
I. Introduction	1
II. System Architecture	2
III. The DS2250 Embedded Controller Chip	4
IV. The Software Package	8
V. Data Acquisition Hardware	12
VI. Example-Using the DPU for Motor Control	12
VII. Concluding Remarks and Areas of Future Work	14
VIII. References	16
Appendix A: Driver Program for Processor Coefficient	28
Appendix B: Digital Low Pass Filter Designs for Signal Processing	37
Appendix C: Start-up Procedures and Hardware Issues	70

I. Introduction

Large space structures are characterized by large order dynamics; typically a subset of these flexible modes are selected as significant parameters but even this set can be on the order of 100 states. Because of the large dimensionality issue, it becomes desirable to process sensor data efficiently and effectively in order to send appropriate information to the controller.

Several researchers have looked at the modelling issues of large space structures [3-7, for example]. The classical approach is to identify the significant flexible modes of the system experimentally and to assume that little, if any, of the spillover effects will significantly reduce the desired closed-loop system response.

Even with a model reduction of the number of flexible modes, there are usually a large number of modes close to the imaging axis that need to be controlled. Compensation requires obtaining sensor information, processing the data efficiently to produce a command signal and transmitting the control strategy through the actuator to the system. With the large spatial and modal structure of the system to be controlled, many space structures process a large number of sensors and actuators.

Sensing [1-2] and the associated pre-conditioning filter problem [8, for example] are important issues that need be addressed efficiently for real-time applications. Typical bandwidth requirements for the significant modes of structure are weighed against sensor and actuator bandwidth as well as inherent noise problems.

Towards this end, this research effort has focused on the development of a hardware tool with associated software and system architecture that addresses some of the requirements for digital signal processing for multi-body flexible structures. The approach is to use a distributed architecture whereby fast single loop (sensor-control-actuator) processing can be accomplished with a supervisor for global coordination. The overall system structure is now enumerated.

II. System Architecture

For a space structure made up of flexible lightweight components, a large amount of data need be processed. Classical architectures use the structure as shown in Figure 1. The entire structure including sensors and actuators is handled as one system; all of the sensor data is sent to a filter, a state estimator if required and a centralized controller. The controller returns a command input to all of the actuators.

The architecture developed here is based upon a distributed configuration as illustrated in Figure 2 and detailed further in Figure 3. Note that in this architecture, the distributed space structure is partitioned according to sensor-actuator loops. Each subsystem then requires local control around the sensor-actuator pair. There are several ways to partition a distributed system into subsystems. One method spatially breaks up the structure into parts, with interaction dynamics between subsystems. Another method partitions the structure according to mode groups where sensors, including filters, separate the frequency spectrum of the structure according to different bandwidths.

Given that some partitioning has occurred, the next design step is to develop the local controller which is implemented for real-time processing. A central processor (supervisor) is used to coordinate global specifications (stability, overall time response characteristics, for example).

There are several filters that can be employed for pre-conditioning of sensor data. Appendix B provides a summary of several designs using the Butterworth, Chebyshev, Elliptic (Cauer) and Bessel filters on sensor data from the CSI Evolutionary Model. For typical space structure system specifications, the following observation can be made about these filters (for a given order): of the structures under study, the Butterworth Filter is the simplest filter to design; the Cauer filter gives the best cut-off characteristics but the worst phase distortion (nonlinearity). Hence the Butterworth and Cauer filters are recommended for further study as signal processors for pre-conditioning.

Table 1.

	Filter type	Passband Characteristics	Cut-off Space	Stop band
1.	Butterworth	Maximally Flat Magnitude Moderate Phase Distortion	Moderate Cut-off	Moderate Attenuation
2.	Chebyshev	Ripples in Passband Extreme Phase Distortion	Sharper Cut-off	Better Attenuation than Butterworth
3.	Cauer	Ripples in passband Extreme Phase Distortion	Sharper Cut-off than Chebyshev	Ripples in Stopband
4.	Bessel	Flat Magnitude. Less Phase Distortion than Butterworth	Slow Cut-off Specifications	Less Attenuation than Butterworth

With the filter design selected, an appropriate controller is required for local control. Both static dissipative control [12] and virtual passive control [13] have been studied. A discrete version of the virtual passive control has also been developed [14]. The controller and filter structure are assumed linear in nature and thus can be described by a transfer function representation or ARMA (auto-regressive, moving average) architecture. With this in mind, the local filter and controller is implemented on an embedded controller, with a higher level computer (minicomputer or 386-based computer, for example) as a supervisor (Figure 4). This controller is based upon the Dallas 2250 chip and associated interface.

The DS2250 Control system is a microcontroller-based, adaptable controller system that can be changed to control a variety of systems. The main idea behind the system is that there is a sensor input, a transfer function representing the processor and an actuator output. The microcontroller can be attached (via a serial cable) to a host computer to allow a variety of transfer functions to be tested.

The basic system contains a DS2250, Intel 8051-based, microcontroller. This controller has 32k of battery backed-up RAM at its disposal, as well as four bi-directional, parallel I/O ports for use with peripherals. The host computer has a C cross-compiler and a download program that works with this micro-controller.

Attached to the DS2250 are an RS-232 interface chip, an Analog to Digital converter, and a Digital to Analog converter. The RS-232 chip is used for communication with the host computer for down-loading and control procedures. The A/D and D/A converters are used to take the signals from the sensor, and send the signals to the system under study (Figure 5).

The software development of this research effort includes several test programs for the converters and the serial port, as well as a library of functions that access the converters. Further, a front-end program has been developed which allows the user to enter the filter/controller characteristics which are then down-loaded to the DS2250 system. Once this is done, control is passed on from the supervisor to the local controllers for local signal processing. These functions are now discussed.

III. The DS2250 Embedded-Controller Chip

The heart of the distributed processor unit (DPU) is the Dallas DS2250 embedded controller (Figure 6). The 40-psi "Soft Micro Stik" is an Intel 8051-based micro controller which has 64K bytes of nonvolatile SRAM for program and data. Using an 8-bit C-MOS baseline, the DS2250 allows the user to change program memory at any time. Serial loading uses an on-chip I/O part to accept data from a host computer via an RS-232 part. This allows a user-friendly program development feature in which the designer can develop and modify code using a personal computer and download pertinent information for control to the DS2250. It is this feature that makes the DS2250 an attractive device for distributed processing development.

Other features of the DS2250 include:

- (1) data RAM or data registers retain information in the absence of a voltage reference,
- (2) extensive security features allows the designs to control unauthorized usage of program RAM,
- (3) the DS2250 is compatible with several existing software and hardware support for the Intel 8051-embedded controller,

- (4) the instruction set also includes interrupts, event counters and timers, serial I/O port capability for synchronous and asynchronous operations and special function registers,
- (5) low power requirements are maintained by power cycling during idle times,
- (6) automatic restart if an errant piece of software was executed; provides the program with robustness to power failure or power surges.

The DS2250 belongs to a family of Dallas embedded controller chips which include the DS5000 (a parallel to the DS2250 but with less RAM), one DS5001Fp (an enhanced version of the DS5000 but with a 80-pin quad flat package) and the DS2256 (the lowest-power version of the DS5001 with 128K bytes of nonvolatile RAM). All of these chips offer flexible read/write memory, SRAM, which is much more attractive than ROM-based controllers. A detail description of this chip now follows.

The Dallas DS2250 (Figure 7) is a family of 8-bit CMOS microcontrollers [10]. The DS2250 is implemented on a 2.65 in by 0.84 in 40-pin SIP. The DS2250 is available in various memory sizes, up to 64K of CMOS RAM, in two maximum clock speeds, 12MHz or 16MHz, and with or without a permanently powered clock/calendar. The RAM memory is nonvolatile, due to an onboard Lithium battery that supplies sufficient power to the RAM for memory retention when power is removed to the system. The DS2250 uses a 8051 instruction set. The DS2250 requires only a crystal and a 5V power supply to become an operational system.

The DS2250 has numerous features that allow for easy implementation of the system. All of the CPU registers are mapped as Special Function Registers (SFR's) and they are identical in number and function to those of the 8051 (i.e., the Accumulator, the Stack Pointer, the B Register, the Program Status Word and the Data Pointer Register). There are 128 scratchpad registers that can be accessed by the CPU. The SFR's are used to establish several other important features of the DS2250.

There is serial I/O capability in the DS2250. The input and output buffers are accessed as a single SFR (SBUF) and control is performed a serial control SFR (SCON). Data that is to be

transmitted is loaded into the SBUF, while received data is read from SBUF. Format of the serial data, such as the number of data bits, the parity bit and the number of stop bits, is set by the control word that was programmed into SCON. The transmit and receive lines for the serial interface use the two least significant bits of the 8-bit parallel port, P3. The band rate is programmed by using one of the programmable timers. The output of the serial I/O port is at CMOS levels (supply voltage is 5V), and thus for compatibility with RS232, these levels need to be converted.

There are two programmable 16-bit timers/counters. Each timer has a SFR for each the high and low byte of the 16-bits. These four SFR can be written to set the timer's intervals, or read to determine the number of events that occurred related to that particular timer/counter. The timers operate by incrementing the timer's 16-bit register each time a pulse is supplied to the timer, providing the timer has been enabled. The pulse can come either from 1/12 of the CPU clock oscillator or from off the board, bits 4 and 5 of parallel port P3. Source of the pulse, timer enabling and mode of timer operation is set by the timer control SFR (TCON). There are various modes in which the timer can operate (i.e., auto-reload, 8 or 16-bit operation) which are set by the contents of TCON. TCON also controls the generation of an interrupt by the timer.

The DS2250 has four 8-bit bidirectional parallel I/O, denoted by P0, P1, P2 and P3. These ports also have other associated functions. P3 has been seen to have two lines associated with serial I/O and two lines associated with the timers. Two more lines of P3 have additional use for external interrupts and the remaining two lines are associated as timing lines for expanded memory. If any of these alternate uses for the data lines are not used, then they revert to parallel I/O. Ports P0 and P2 have the alternate use of forming a 16-bit data bus for expanded memory (off SIP memory). If the expanded memory is not implemented P0 and P2 remain as parallel I/O ports. All of the ports have pull ups on each data line. To output data, the data is written to the desired port. While for input, a 0xFF is first written to the port and then the port is read.

There are various configuration parameters that are protected from errant modification. Some of these are the memory partition data bits, the security lock bit, the watchdog timer enable

and reset bits, the stop mode bit and Power-On-Reset bit. To protect this data, a timed access method is used to alter any of these data bits. To modify any of these data bits, a 0xAA is written to the Timed Access Register followed within two machine cycles of a write of 0x55 to the Timed Access Register. The next instruction can alter any of these configuration parameters, within the next two machine cycles. If the 0x55 is not written within the required time following the writing of 0xAA, or the timed access restricted data is not modified within its time limit of the writing of 0x55, then the modification of the data is not permitted. This method is used to prevent errant code from destroying the machine configuration. The logic is that performing the two writes and the modification of the configuration in sequence and in the time limit is highly unlikely by errant code (i.e., pure chance). Information contain in these configuration parameters can be read without going through the timed access routine, but they can not be altered with following the routine exactly.

There is circuitry on the DS2250 to detect a power fail condition. If enabled this will generate an interrupt. This will allow for the storing of the CPU state in memory for retrieval during a power up. With the power off the system will retain its memory contents because of the Lithium battery. This allows for the program to resume operations after the power failure.

The DS2250 also has a Watchdog Timer. This is used to reset the DS2250 in the event that software control is lost. The Watchdog Timer is enabled via a Timed Access setting of the Watchdog Timer Enable. The timeout period is equal to 122880 machine cycles. The Reset Watchdog Timer (RWT) is set high using a Timed Access routine. The RWT must be set high before the Watchdog Timer times out, as long as the Watchdog Timer is enabled, otherwise the Watchdog Timer generates a reset signal for the DS2250. The Watchdog Timer provides for a reset the DS2250 gets into a "Death Grip" with some device, or any situation occurs in which the software is unable to reset the timer within the allotted time.

The DS2250 also has a Security Lock, Encrypting Logic and internal Vector Ram for interrupt servicing. The Security Lock provides a lock on the program memory and configuration data. The lock is set at the completion of programming, if desired. The program memory can be

read only if the Security Lock is reset. Once the Security Lock is set, the program memory can not be read and if the Security Lock is then unlocked the contents of the CPU and memory are zeroed. The CPU can still operate from the program in memory when the system is locked. At the time of programming the DS2250, the program can be encrypted. The encryption routine is based on the use of a 40-bit encryption word that is set during programming. The encryption word is used to encode both the program code, data and addresses used by the system. The internal Vector RAM is used to by the system during encrypted operation to prevent sending of addresses or data to the memory chips on the DS2250 SIP during an interrupt service routine or during a power on reset.

The DS2250 has no ROM for program storage. All program and data memory is in the system's RAM, and since the memory is nonvolatile due to the Lithium battery it is not lost even when the system's power is removed. This allows for the changing of the memory contents without the need to erase an EPROM or changing to a new DS2250. The memory can be programmed either through a serial or parallel interface. During programming, in addition to the loading of the program code, the system's memory partition is set, the system can be locked and the encryption key word can be set. Parallel programming of the DS2250 has not been perused, due to the simplicity in hardware and software available for serial programming.

IV. The Software Package

There is a programming package available from Dallas Semiconductor, called KIT5K [10]. KIT5K runs on an IBM PC/AT compatible machine operating under DOS2.0 or higher. This software communicates via either of the COM ports on the machine to the DS2250. Since the COM port is RS232 and the DS2250 requires CMOS logic levels, a RS232 to CMOS converter is required. The KIT5K software toggles the DTR line of the COM port which can be connected to the Reset and PSEN* line of the DS2250 to put the system in the serial programming mode, or this can be performed manually (if desired). The DTR must also be converted from RS232 to CMOS levels. Once the Reset and PSEN* lines are activated to assert serial programming mode, the DS2250 senses the serial port receiver line to detect the <CR> transmitted by the programming

computer. From this serially transmitted character, the DS2250 is able to determine the baud rate at which it is to be programmed and the serial programming mode is established. Not all baud rates are available. The usable standard baud rates for various crystal frequencies are published in the User's Manual DS5000.

There are numerous commands for the KIT5K software, and all of them will not be examined here. Most of the commands can be placed into three categories: PC Control, Programming Control and Memory Editing. Some of the commands from each category will be examined.

The PC Control category contains commands primarily used to control the PC that KIT5K is operating on. The COM command directs through which port the KIT5K software will communicate with the DS2250. The SPEED command sets the baud rate and DTR toggles the DTR line, of the selected com port. Since on each entry into KIT5K requires that the com port and the baud rate to be set, to avoid exiting KIT5K there is a DOS command that allows DOS command execution without leaving KIT5K. In addition, there are some DOS commands built into the KIT5K to allow changing the current directory or listing its contents.

The Programming Control commands are to facilitate programming of the DS2250. Obviously, the dominant command is PROGRAM which performs all the steps required to program the DS2250 with a specified Intel hex format file containing the program code. There are individual commands to perform each item required to program the DS2250 from the PC. The PROGRAM command automatically performs the required items and queries the user about programming options. Once KIT5K has been initialized, PROGRAM along with the specified hex file is used as a command. KIT5K then executes a DTR command to initialize serial programming and then a <CR> is transmitted to establish serial programming. An UNLOCK command is executed to unlock the DS2250 Security Lock. The software then queries the user for configuration data, namely memory size, partition address, encryption and key if needed, beginning address, ending address and if the programming is to be verified. After these values have been entered, the PROGRAM command then executes the various commands needed to

program the DS2250 as specified. The RANGE command is used to set the memory size of the DS2250. The PARTITION command is executed to set the specified address for the program/data space partition. If encryption is being used the KEY command is executed to load the DS2250 with the encryption key. The hex format file is then downloaded to the DS2250 via the LOAD command. If verification of the downloaded program is required the VERIFY command is executed to compare the hex file with the memory contents. If the system is to be locked, a LOCK command is executed to assert the DS2250's Security Lock. And finally, the RUN command is executed which places the DS2250 in the execution mode. In order to simplify the process a batch and configuration file can be used to perform these tasks. The batch file delivers the commands to be executed, while the configuration file has the system configuration data in it.

The Memory Editing commands allow for the altering and examination of the DS2250's memory. These commands are similar to the commands of DDT or DEBUG. The FILL command fills the specified memory locations with the specified constant. The EDIT command permits the user to sequentially view and alter memory starting at the specified address. DUMP stores the contents of the DS2250 memory between the two specified address in a specified file. The UNASM command unassembles the contents of the DS2250's memory.

There are numerous ways of generating an Intel hex format file containing the program code. There are many assemblers and cross compilers that run under DOS that produce these files. The Franklin 'C' Compiler [11] does meet this requirements and operates under DOS3.0 or higher. This 'C' compiler and its associated programs generate Intel hex format files for the 8051 family of microcontrollers and related microcontrollers. The compiler package is composed of the 'C' compiler, an assembler, a linker, a library manager, and an object to hex converter.

The 'C' compiler implements an ANSI standard 'C'. There are extensions to this package to allow for easy access to the microcontroller's hardware. This is accomplished to two ways. First, the specific Special Function Registers are defined in a CPU specific header file that contains the names of the SFR/s mnemonic as a variable. Thus, an output of 0xF8 by parallel port 2 is achieved by the statement, P2=0xF8;. One of the extensions of the compiler allows for bit

communications with the SFR's mnemonics followed by a $\wedge i$, with i being the desired bit number. To reset the third bit of port 2, use, $P2 \wedge 3=0$;. Operations with the SFR's or particular bits of the SFR's is easily performed in the 'C' program. In addition particular bits can be given unique names by the define statement. The 'C' program can access all of the DS2250, without the user reverting to assembler code. It should be noted that a timed access write and interrupt servicing can be performed in 'C' code.

The Franklin 'C' compiler, while it can easily communicate with the DS2250's hardware, does not permit for the inclusion of assembler code inside the 'C' code. Items that need to be executed in assembler must be written as an assembler function, assembled and linked to the object code generated by the compiler. In addition, the compiler can be linked to functions generated in Intel PL/M 51. The output of the linker has a file extension of abs. This absolute object file may contain symbol information for symbolic debugging. The object-hex-symbol converter generates an Intel hex format file from the output of the linker. This hex file is then programmed into the DS2250 by the KIT5K software.

Besides these system files, a batch file has been developed as a user-friendly communication between the designer and the DS2250 system. This file, tf.bat, is resident on the supervisor system (IBM system in this study) and allows the user to create a continuous-time transfer function or a discrete-time (digital) transfer function. In the continuous-time case, a discrete-time equivalent transfer function is created using the bilinear z-transformation. Currently, the system allows for a fifth-order continuous-time system or a fifteen-order discrete-time system. These parameters can easily be changed by an appropriate modification in the driver software.

Once the local processor coefficients are entered through the supervisor system, it is downloaded to the DS2250 embedded controller. From this point on, the local controller takes over. Through the use of data acquisition chips, the local processor obtains information from the sensor and sends commanded inputs to the actuator.

Note that the transfer function form has applicability to many processing systems. The filters designed in Appendix B, for example, can be readily tested using this local processor.

Further, classical PID (proportional, integral, derivative) control, phase compensation and virtual passive control can be implemented on this system.

Issues such as bandwidth limitations, system order, voltage level constraints and software requirements are presented in Appendix C.

V. Data Acquisition Hardware

The analog-to-digital converter is the Analog Devices ADC0804C chip. This system has 8-bit resolution with differential inputs and a sampling rate of up to 10KHz. The converter has a reference voltage of 5v and operates with a 256-resistor DAC network.

The digital-to-analog converter is the Analog Devices AD7530/533. This system is a 10-bit multiplying DAC using CMOS technology and uses a 5v reference (it can operate with voltage references up to 15 volts). An inverted R-2R ladder is used for the binary weighed operations; a sampling rate is typically quoted at 160ns (6.7MHz). The DAC used in this study is set for bipolar operation (4-quadrant multiplication).

Finally, an RS-232 interface circuit (the ICL232CPE) is used to interface the Dallas DS2250 controller to an IBM computer. Pin connections are shown in Figure 8. All of the data acquisition components are fully functional and are transparent to the user. Further software to analyze the data acquisition units and to transfer data and command signals back and forth between the embedded controller and the data acquisition system have been fully tested; upon start up, these modules are activated and the user need not access any additional files.

A discussion of the start up procedure and hardware issues is provided in Appendix C.

VI. Example-Using the DPU for Motor Control

A two-beam flexible testbed has been constructed for collocated and non-collocated control studies. A separate report [15] summarizes some of the initial studies on non-collocated control using zero placement and phase compensation. The testbed has a stepper motor produced by Servo Systems which rotates along the z-axis (vertical). Two flexible steel alloy beams are

attached to a mounting unit. These beams have been selected so that all movement is in the plane parallel to the ground; little torsional motion is generated, as desired.

Two servo motors (produced by the Maxon Company) provide actuation for these beams. These motors were selected because of the speed requirements to suppress vibration. Two Copley control servo amplifiers provide the necessary voltage from the computer to the motor. Each motor has an incremental optical encoder (produced by Hewlett-Packard) with 1024 cycles per resolution.

Each beam has a 350 ohm strain gauge mounted along a side which sends non-collocated position information through an amplifier circuit to the data acquisition system. Figure 9 illustrates the two-beam flexible testbed.

For the purpose of demonstrating the DPU, the two-beam unit was replaced by a single beam with a different mounting unit. The objective of the test was to select controller gains to suppress beam vibration. Proportional-derivative control was selected for illustration. Files download.bat 9.1, 9.2, 9.3 contain the data for these trials. File 9.1 contains the proportional gain, 9.2 contains the derivative gain and 9.3 contains the desired terminal angle in radians.

First, the user loads the start-up file tf.bat. Then the software prompts the user for controller/filter parameters and upon completion of the data entry, downloads the data and passes control to the DPU.

Figure 10 illustrates the results for four different runs. For this single study, one observes that a proportional gain of -10 and a derivative gain of +8 provides a desirable result for vibration suppression. The plots are in seconds and show the output of the strain-gage, i.e., the position of the beam during the movement from 0° to 45°.

Other tests have been performed on the DPU and it is noted that this embedded controller system is a viable architecture for distributed signal processing application for flexible structures such as those currently being considered in spacecraft designs.

Bandwidth requirements for most applications under study are satisfied with the 10KHz data acquisition unit (a higher bandwidth can be easily added on by changing the DAC and

adjusting the clock for the A/D); further adequate accuracy is achieved into the 8-10 bit data conversion and 8-bit microcontroller.

VII. Concluding Remarks and Areas of Future Work

This research effort has resulted in an embedded controller system which can be used as part of a distributed signal processing architecture for flexible multi-body structures currently under investigation at NASA. The distributed processing unit (DPU) is a self-contained processor which can be attached to a supervisory computer through an RS-232 port and to a sensor-actuator pair through a data acquisition system. Accuracy and sampling rates are satisfied for many multi-body structures currently being used at the Spacecraft Dynamics Branch.

Areas of future activities are many. First of all, microcontroller chips are constantly being upgraded and many of the new designs are smaller and require less power. For example, the Dallas DS2255 which is currently under design/fabrication is a single 68-Pin SIP stick data acquisition system (A/D and D/A). This chip has 12-bit A/D accuracy, 8-channel input, has a digitally controlled channel selector and a 50KHz maximum throughput rate. There are 8 analog outputs and an 8-bit output resolution. Further this chip has an addressable serial interface, requires just a +5 reference voltage and an external A/D clock. This chip would reduce the size of the DPU.

The DS2255 instrument stick could be interfaced with the DS2256 power miser micro stick - an embedded controller chip also currently under design and fabrication. This 68-Pin SIP stick is an extremely low power chip (+3v to +5v reference) which includes 128K bytes of nonvolatile SCRAM, a watchdog timer, a permanently powered timekeeper, has 29 bits of user-defined I/O ports and a kickstarter if an interrupt or idle status was previously enabled.

Once these chips are available, the existing DPU developed in this study can be directly transferred to upgrade the hardware technology. Speed and com compactness would thus be upgraded.

Another area of future work is in the area of filter designs. Classical algorithms have been employed for the pre-conditioning processors. Thus a transfer function model (or ARMA structure) has been implemented in the user-friendly driver package. Other filter designs such as bandpass, notch and all-pass structures as well as adaptive, self-tuning systems may be developed. This would enhance the capabilities of the DPU when the model of the structure under study is unknown or certain parameters are uncertain. Further the filter design can be enhanced to work with the controller under study. To do this, many controller structures allow some flexibility in zero-placement or eigenvector selection. The filter component not only has the requirement to suppress certain frequency components but also has the disadvantage of adding phase distortion or time delays. The controller has to compensate for some of these problems.

Finally, distributed processing systems requires some protocols between the supervisor and local controllers. Approaches from computer sciences (queuing, token ring structures) and optimal control theory (min-max performance measures, game theory) may be employed to handle priorities of data sharing between local controllers when necessary.

These areas of future research would benefit the current objectives of the control/structure interaction program by providing an efficient control architecture for multi-body flexible structures.

VIII. Reference

1. Sievers, L. Blackwood, G. and Von Flotow, A., "MIMO Narrowband Frequency Shaping with CSI Application", Proc. of the 1991 American Control Conference, Boston, 1991.
2. Montgomery, R. and Ghosh, D., "Distributed Sensing in Control of Large Flexible Spacecraft", Proc. of the 1991 American Control Conference, Boston, 1991.
3. Yam, Y., Bayard, D., and Scheid, R., "Frequency Domain Identification for Robust Large Space Structures Control Design", Proc. of the 1991 American Control Conference, Boston, 1991.
4. Tseng, D., Lew, J., Longman, R., and Juang, J. N., "Comparison of ERA, ERA/DC and Q-Markov Identification Algorithm Based on Variance and Bias", Proc. of the 1991 American Control Conference, Boston, 1991.
5. Banavar, S., Jean-Noël, A., and Kenneth, R. L., "Identification Experiment for Control of a Flexible Structure", IEEE Control System Magazine, May 1985.
6. Darns, J. A. and Eugene, M. C., "On the Formulation of Some Distributed System Parameter Identification Problems", Proc. Symp. Dynamics and Control of Large Flexible Structures, Blacksburg, VA, 1977.
7. Banks, H. T., Gates, S. S., Rosen, I. G., and Wang, Y., "Identification of a Distributed Parameter Model for a Flexible Structure", *SIAM J. of Control Opt.*, October 1986.
8. Wie, B., Liu, Q., and Byun, K. H., "Robust H-infinity Control Synthesis Method and Its Application to a Benchmark Problem", Proc. of the 1990 American Control Conference, San Diego, 1990.
9. Haykin, S., Adaptive Filter Theory, Prentice-Hall, New Jersey, 1986.
10. Dallas Semiconductor, DS5000 Soft Microcontroller User's Guide, Jan. 1990.
11. C51 Compiler Reference Manual, Ver. 10.90, Franklin Software, Inc., ©1990.
12. Joshi, S. M., Maghami, P. G., and Kelkar, A. G., "Dynamic Dissipative Compensator Design for Large Space Structures", AIAA Guidance, Navigation and Control Conference, New Orleans, 1991.
13. Juang, J. N. and Phan, M., "Robust Controller Designs for Second-Order Dynamic Systems: A Virtual Passive Approach", NASA TM-10266C, Langley Research Center, Hampton, VA, 1990.
14. Nimmo, N. and Lee, G. K. F., "Low Pass Filter Designs for the CSI Test Article: Numerical Simulation Results", submitted to the ISMM Conf. on Computer Applications in Design, Simulation and Analysis, Orlando, 1991.
15. Lee, G. K. F., Strand, S., and Bird, W., "A Two-Beam Flexible Testbed for Non-Collocated Control Studies", ISMM Conference on Computer Applications in Design, Simulation and Analysis, Las Vegas, 1991.

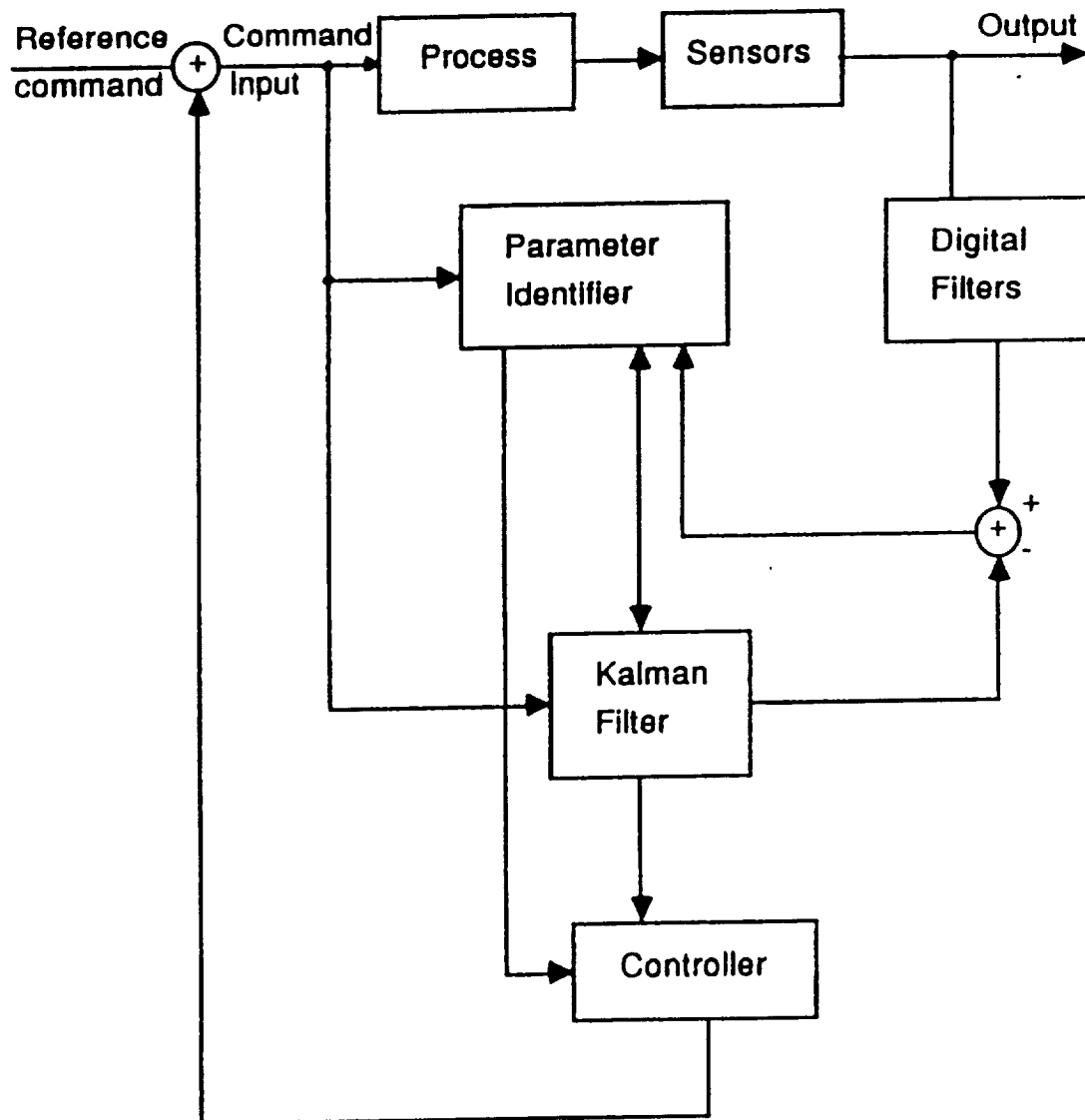


Figure 1: Overall system with estimator and control blocks

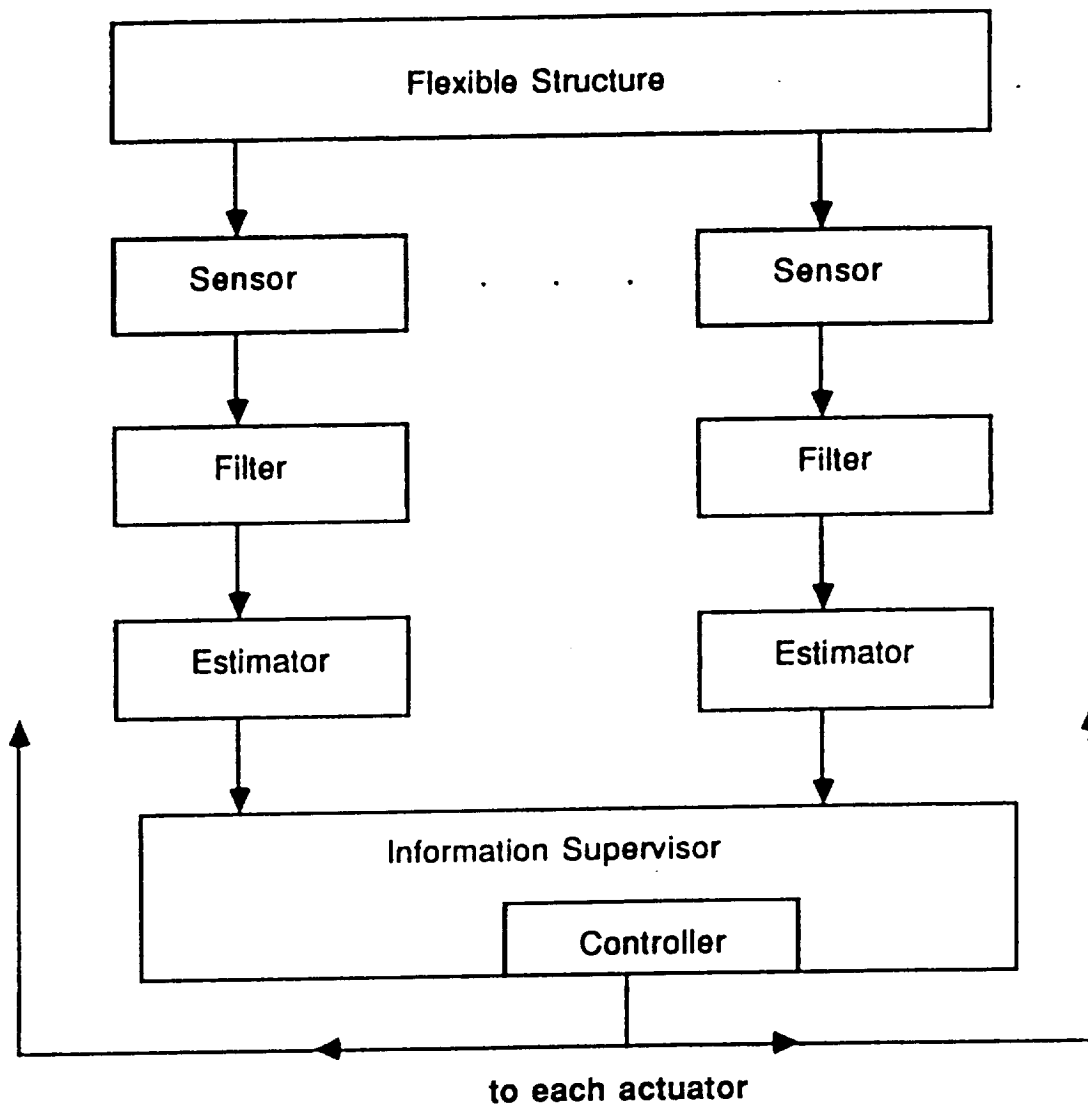


Figure 2: Distributed processor architecture

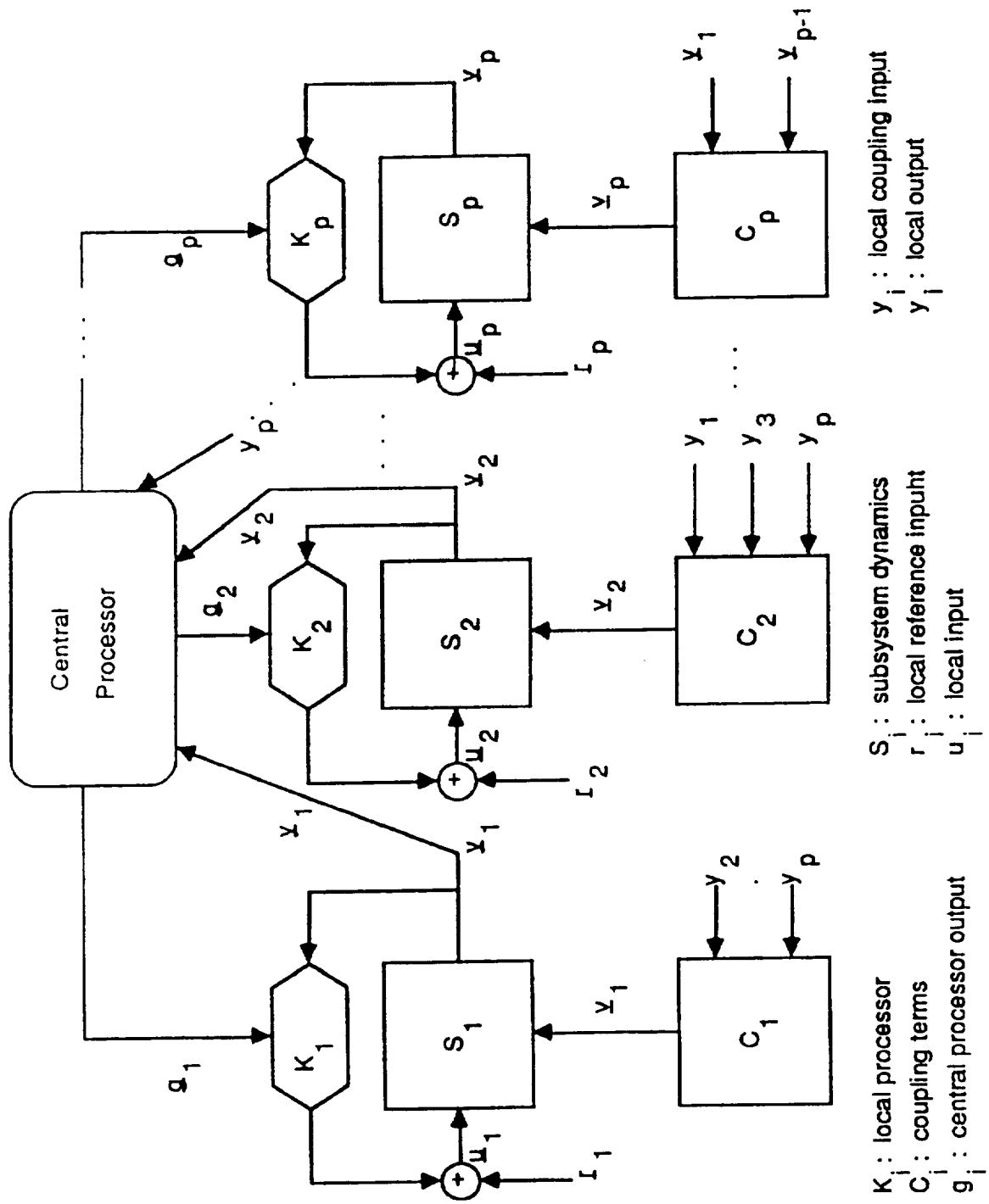


Figure 3: A large-scale distributed system

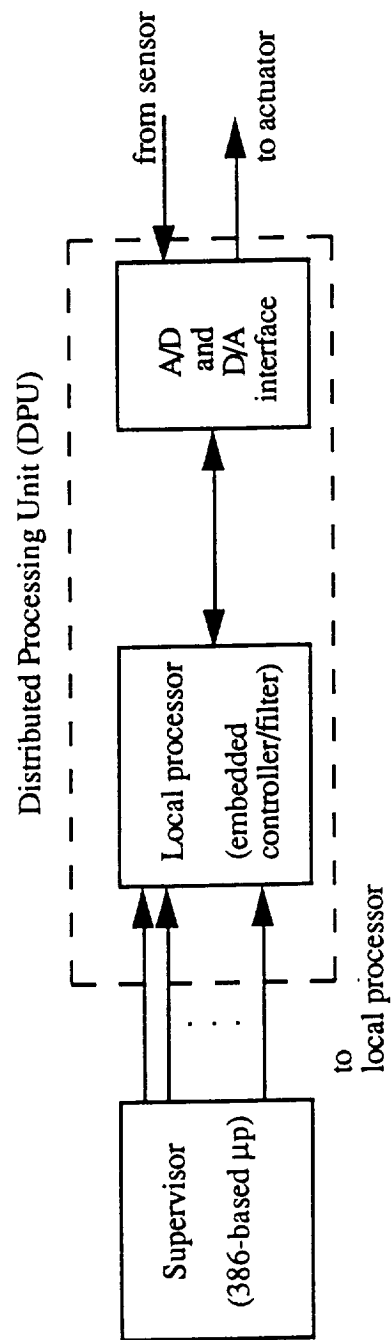


Figure 4: System architecture in block form

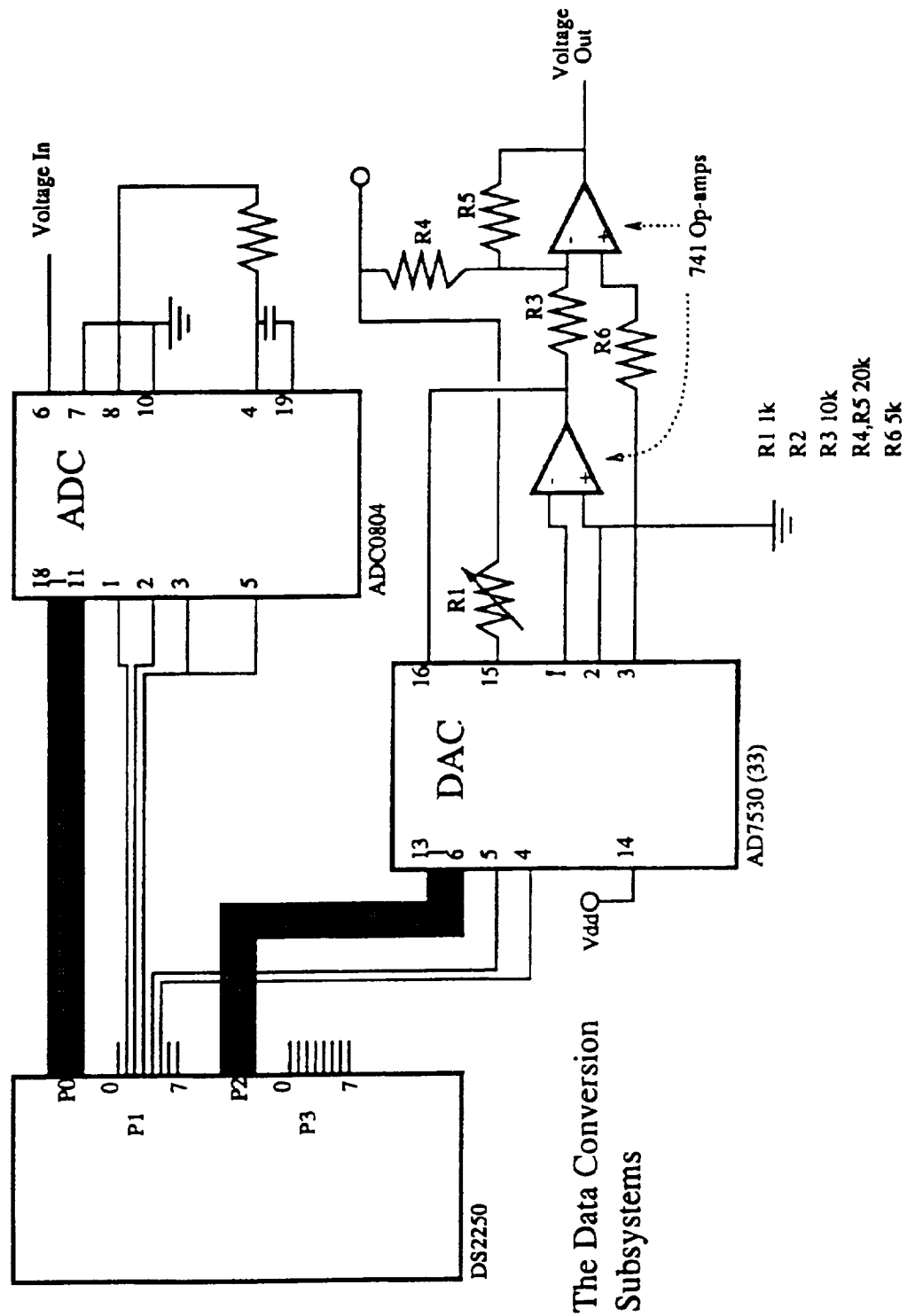
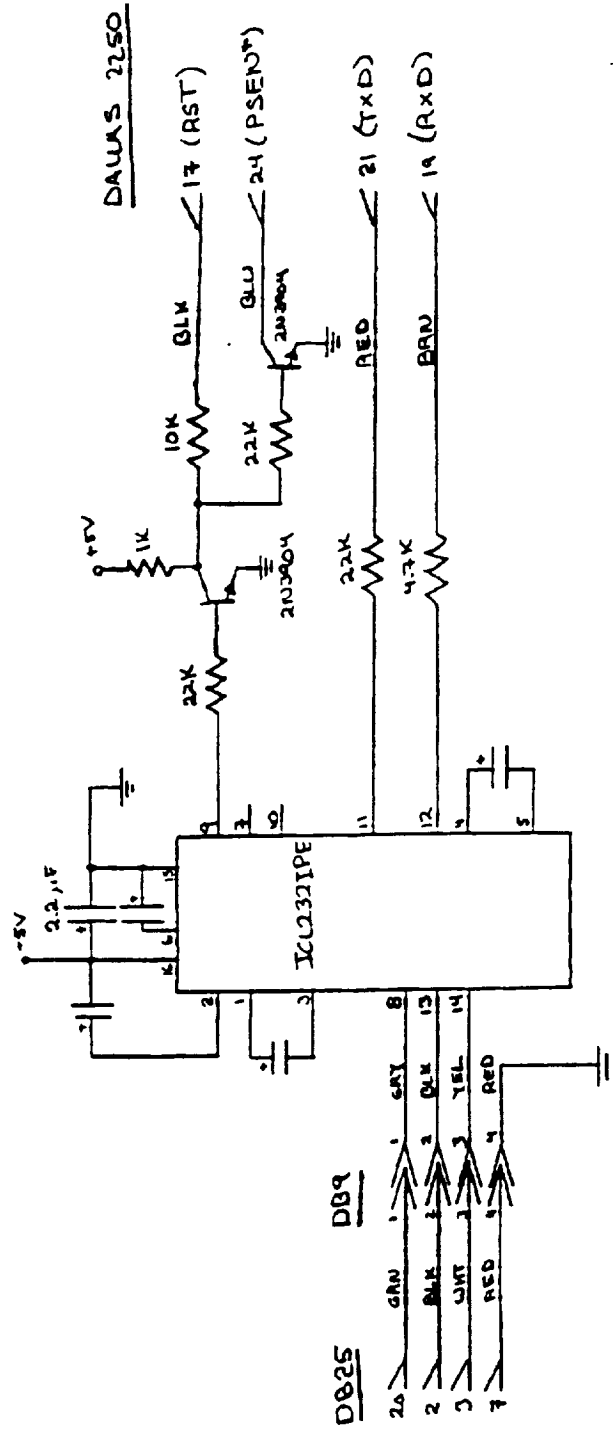
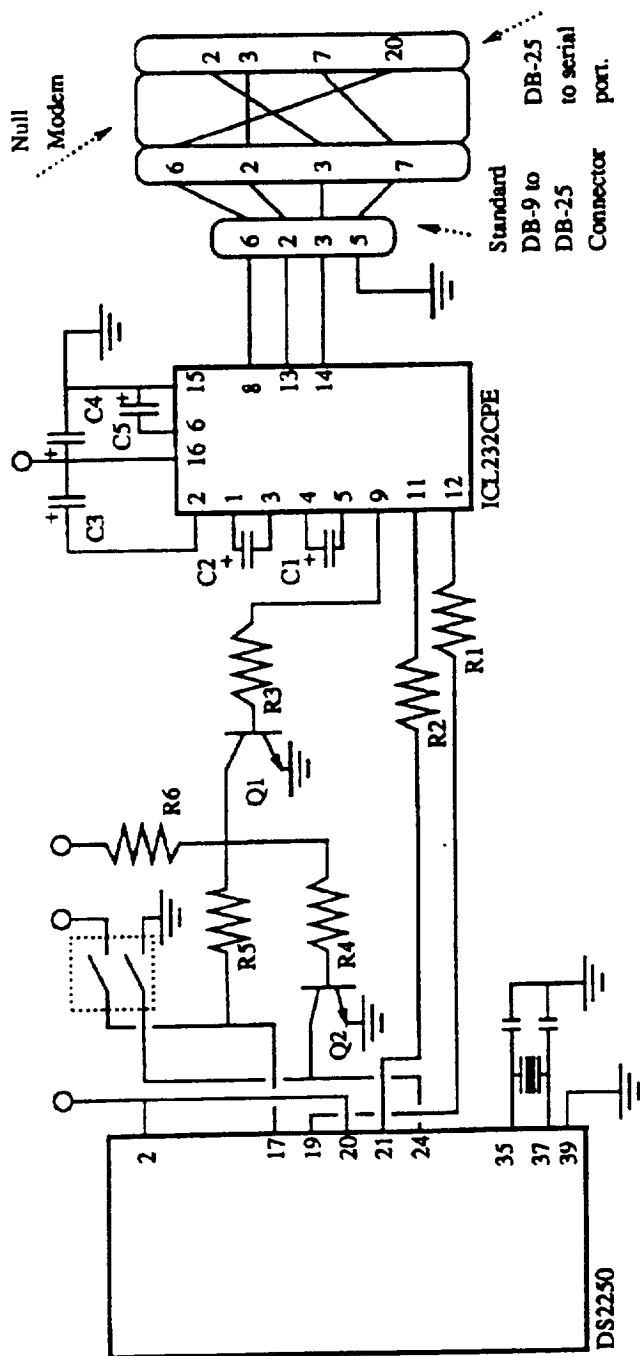


Figure 5: The hardware architecture



Note: All capacitors are 10µF, 25V
Tantalums, unless noted

Figure 6: The Dallas 2250 hardware design



Basic DS2250 system

R1: 4.7K	Q1, Q2: 2N3904	C1-C5: 10nF, 25V
R2-R4: 22K		
R5: 10K		
R6: 1K		

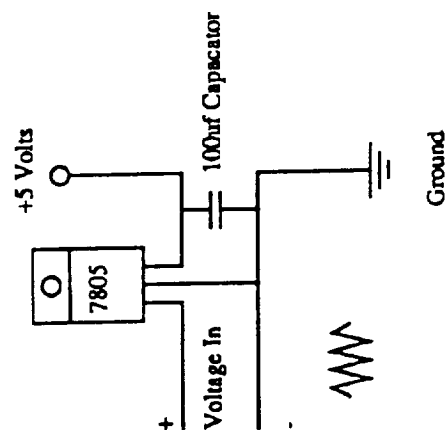


Figure 7: Basic 2250 showing per-connector

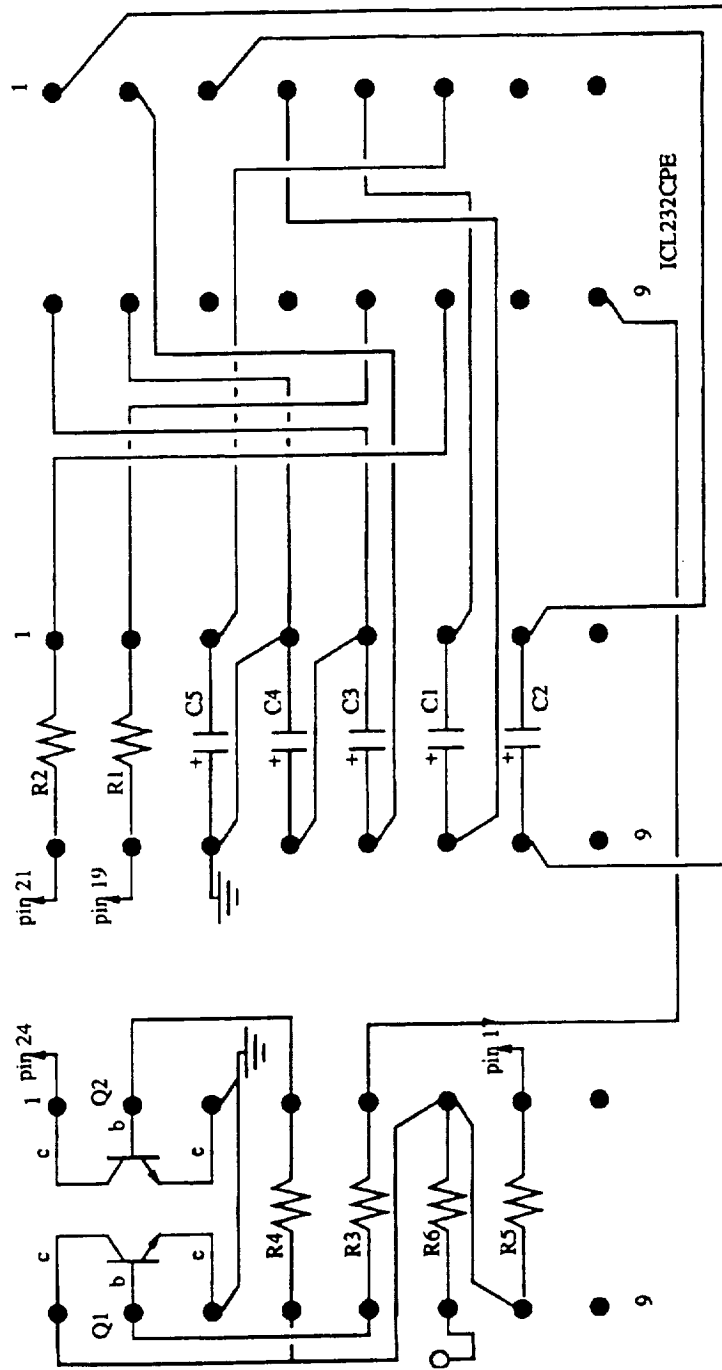


Figure 8: Bottom view of RS-232 interface components

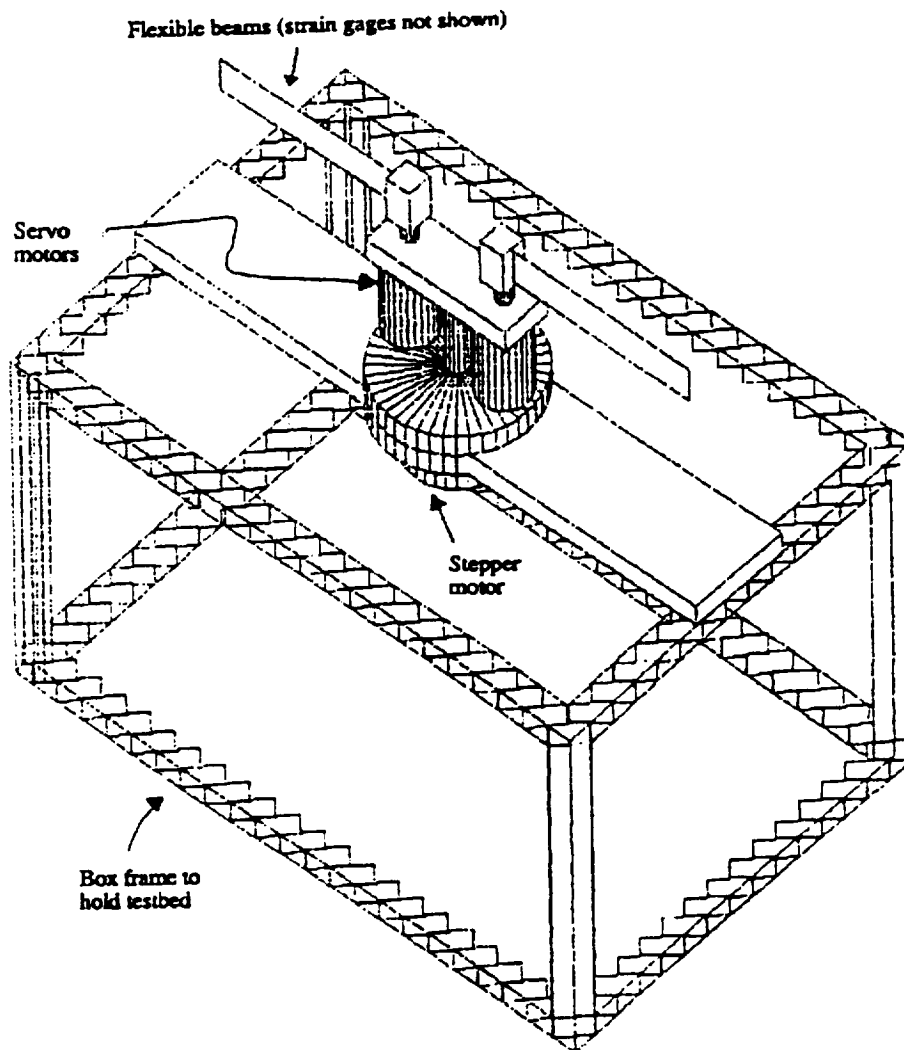


Figure 9: Two-beam three-axis flexible testbed

hp stopped

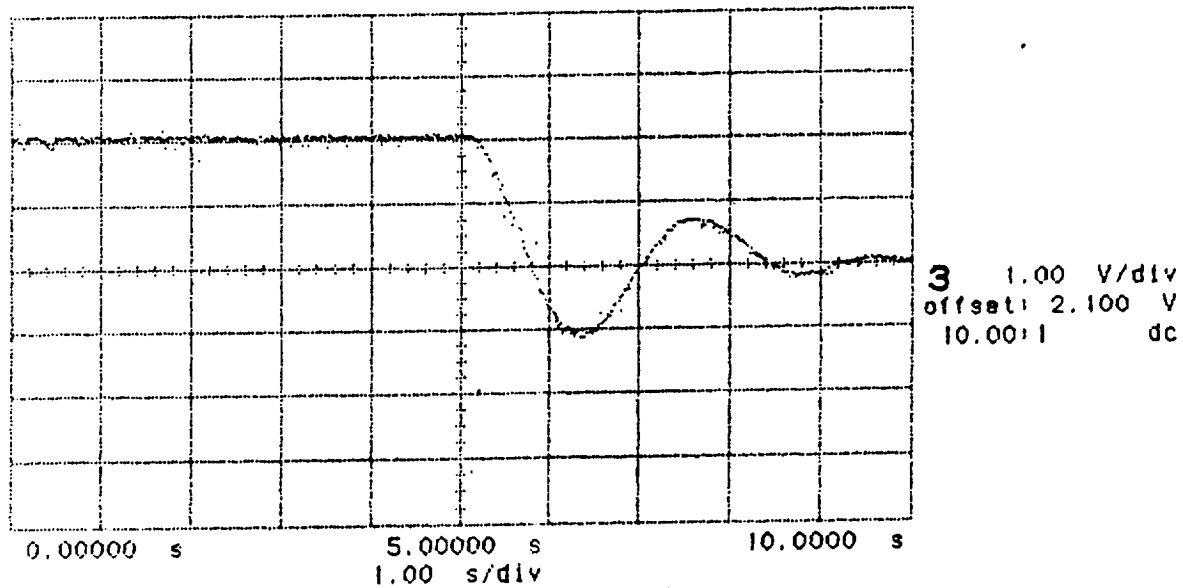


Figure 10a: $P = -1$, $D = -0.1$

hp stopped

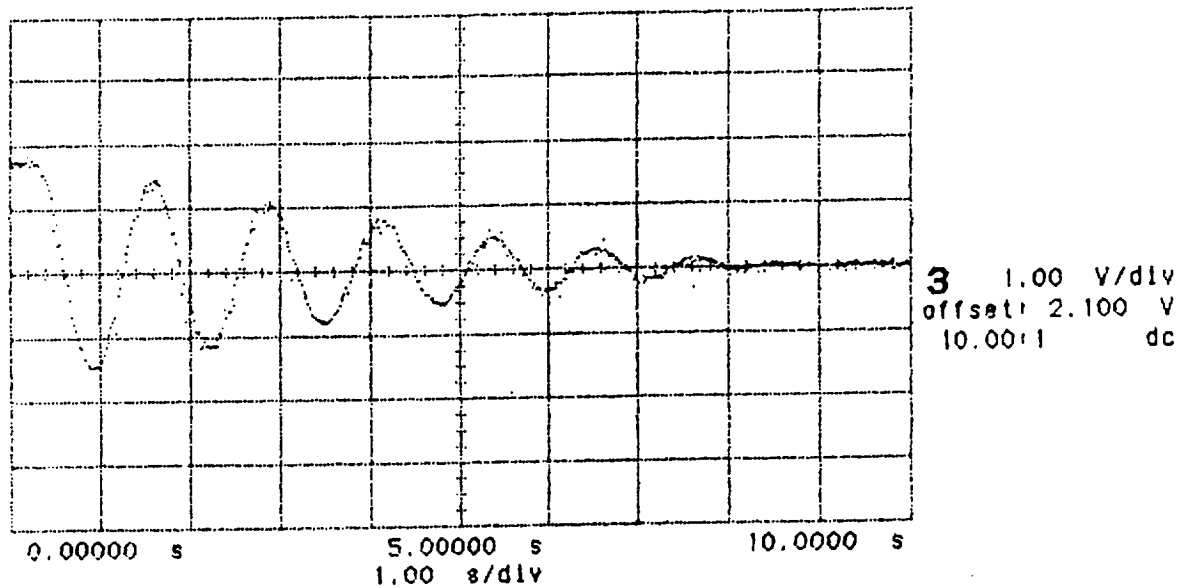


Figure 10b: $P = -4$, $D = -0.2$

hp stopped

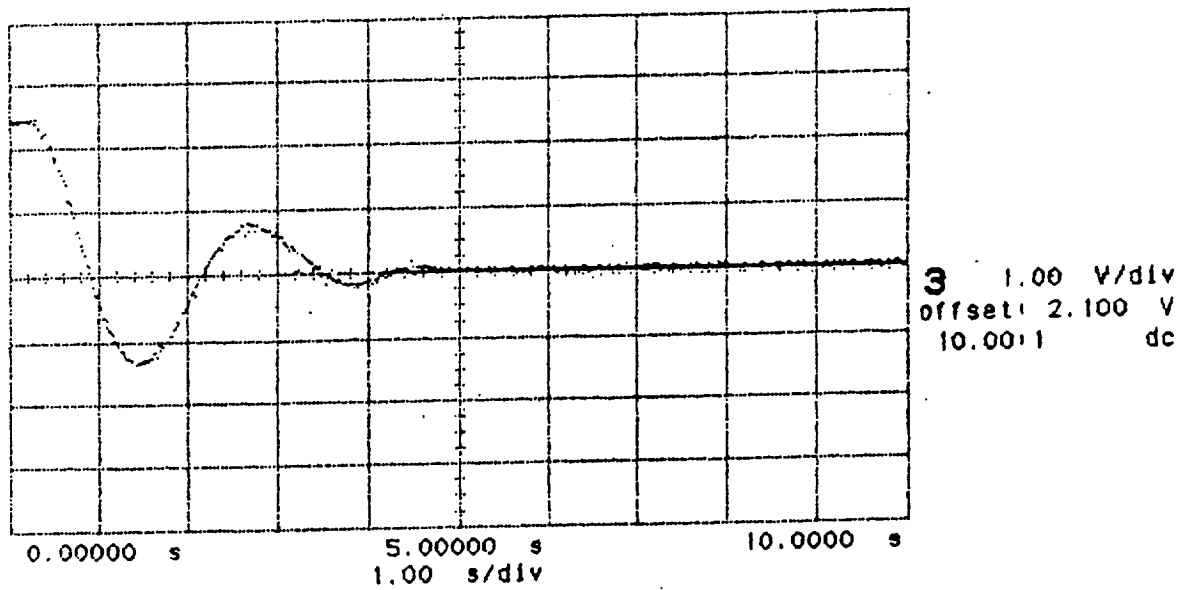


Figure 10c: $P = -4$, $D = 2$

hp stopped

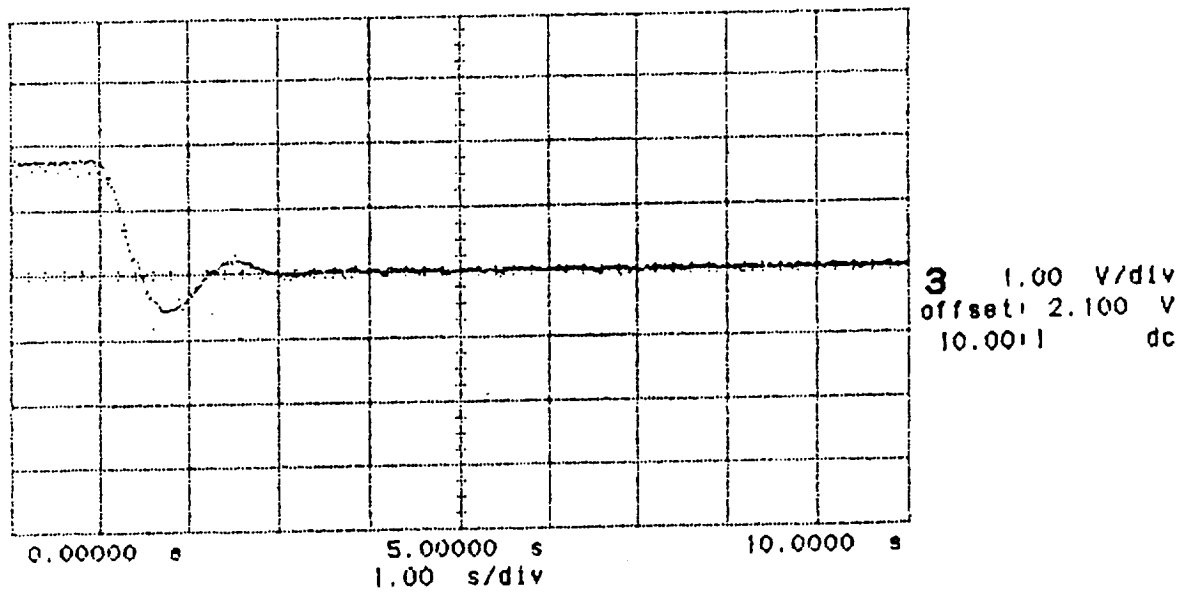


Figure 10d: $P = -10$, $D = 8$

Appendix A
Driver Program for Processor Coefficients

```

A:\> a
-----
10 REM-----
20 REM      Andrew Meyers   Fall 1991
30 REM
40 REM      This program gets the coefficients either of a continuous-
50 REM      time transfer function or a discrete-time transfer function
60 REM      from a user. If a continuous-time transfer function is entered
70 REM      it is converted to discrete-time. When all of this is complete
80 REM      the function is downloaded to the microcontroller and the motor
90 REM      sequence is begun.
-----
100 REM-----
110 CLS
120 KEY OFF
130 LOCATE 8,20:PRINT"MOTOR CONTROL TRANSFER FUNCTION"
140 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
150 PRINT:PRINT
160 PRINT"FOR CONTINUOUS-TIME INPUT A 1"
170 PRINT"FOR DISCRETE-TIME INPUT A 2"
180 INPUT CD
190 IF CD =1 OR CD =2 THEN GOTO 220
200 PRINT:PRINT"PLEASE INPUT A 1 OR A 2 ONLY"
210 GOTO 150
220 DIM A(10),B(10),AZ(30),BZ(30)
230 FOR I=0 TO 10
240 A(I)=0
250 B(I)=0
255 NEXT I
256 FOR I=0 TO 30
260 AZ(I)=0
270 BZ(I)=0
280 NEXT I
290 IF CD=2 THEN 1410
-----
300 REM-----
310 CLS
320 PRINT"CONTINUOUS-TIME TRANSFER FUNCTION"
330 PRINT:PRINT
340 INPUT"Input degree of Numerator Polynomial";ND
350 IF ND<=5 THEN 380
360 PRINT:PRINT"The order of the polynomial must be <= 5"
370 GOTO 330
380 IF ABS(INT(ND))<>ND THEN 330
390 PRINT:PRINT
400 PRINT"Numerator coefficients are assumed to be of the form:"
410 PRINT
420 PRINT" $A(n)*s^n+A(n-1)*s^{(n-1)}+...+A(1)*s+A(0)$ "
430 PRINT:PRINT
440 FOR I=ND TO 0 STEP -1
450 PRINT"Input coefficient A(";I;")"
460 INPUT A(I)
470 NEXT I
480 PRINT:PRINT
490 INPUT"Input the degree of Denominator Polynomial";DD
500 IF DD<=5 THEN 530
510 PRINT:PRINT"The order of the polynomial must be <= 5"
520 GOTO 480
530 IF ABS(INT(DD))<>DD THEN 480
540 PRINT:PRINT
550 PRINT"Denominator coefficients are assumed to be of the form:"

```



```

560 PRINT
570 PRINT "B(n)*s^n+B(n-1)*s^(n-1)+...+B(1)*s+B(0)"
580 PRINT:PRINT
590 FOR I=DD TO 0 STEP -1
600 PRINT "Input coefficient B(";I;")"
610 INPUT B(I)
620 NEXT I
630 REM-----
640 CLS
650 PRINT:PRINT "DIGITAL CONVERSION WILL NOW TAKE PLACE"
660 PRINT "Input the sampling period (sec) of the system"
670 INPUT T
680 C=2/T
690 IF DD=0 THEN 760
700 IF DD=1 THEN 790
710 IF DD=2 THEN 870
720 IF DD=3 THEN 970
730 IF DD=4 THEN 1090
740 IF DD=5 THEN 1230
750 REM-----
760 TF=A(0)/B(0)
761 IF DD=0 THEN AZ(0)=TF
770 GOTO 1380
780 REM-----
790 REM    First Order bilinear transformation
800 REM
810 AZZ=B(0)+B(1)*C
820 AZ(0)=(A(0)+A(1)*C)/AZZ
830 AZ(1)=(A(0)-A(1)*C)/AZZ
840 BZ(1)=(B(0)-B(1)*C)/AZZ
850 GOTO 1380
860 REM-----
870 REM    Second Order bilinear transformation
880 REM
890 AZZ=B(0)+B(1)*C+B(2)*C^2
900 AZ(0)=(A(0)+A(1)*C+A(2)*C^2)/AZZ
910 AZ(1)=(2*A(0)-2*A(2)*C^2)/AZZ
920 AZ(2)=(A(0)-A(1)*C+A(2)*C^2)/AZZ
930 BZ(1)=(2*B(0)-2*B(2)*C^2)/AZZ
940 BZ(2)=(B(0)-B(1)*C+B(2)*C^2)/AZZ
950 GOTO 1380
960 REM-----
970 REM    Third order bilinear transformation
980 REM
990 AZZ=B(0)+B(1)*C+B(2)*C^2+B(3)*C^3
1000 AZ(0)=(A(0)+A(1)*C+A(2)*C^2+A(3)*C^3)/AZZ
1010 AZ(1)=(3*A(0)+A(1)*C-A(2)*C^2-3*A(3)*C^3)/AZZ
1020 AZ(2)=(3*A(0)-A(1)*C-A(2)*C^2+3*A(3)*C^3)/AZZ
1030 AZ(3)=(A(0)-A(1)*C+A(2)*C^2-A(3)*C^3)/AZZ
1040 BZ(1)=(3*B(0)+B(1)*C-B(2)*C^2-3*B(3)*C^3)/AZZ
1050 BZ(2)=(3*B(0)-B(1)*C-B(2)*C^2+3*B(3)*C^3)/AZZ
1060 BZ(3)=(B(0)-B(1)*C+B(2)*C^2-B(3)*C^3)/AZZ
1070 GOTO 1380
1080 REM-----
1090 REM    Fourth Order bilinear transformation
1100 REM
1110 AZZ=B(0)+B(1)*C+B(2)*C^2+B(3)*C^3+B(4)*C^4
1120 AZ(0)=(A(0)+A(1)*C+A(2)*C^2+A(3)*C^3+A(4)*C^4)/AZZ
1130 AZ(1)=(4*A(0)+2*A(1)*C-2*A(3)*C^3-4*A(4)*C^4)/AZZ
1140 AZ(2)=(6*A(0)-2*A(2)*C^2+6*A(4)*C^4)/AZZ

```

```

1150 AZ(3)=(4*A(0)-2*A(1)*C+2*A(3)*C^3-4*A(4)*C^4)/AZZ
1160 AZ(4)=(A(0)-A(1)*C+A(2)*C^2-A(3)*C^3+A(4)*C^4)/AZZ
1170 BZ(1)=(4*B(0)+2*B(1)*C-2*B(3)*C^3-4*B(4)*C^4)/AZZ
1180 BZ(2)=(6*B(0)-2*B(2)*C^2+6*B(4)*C^4)/AZZ
1190 BZ(3)=(4*B(0)-2*B(1)*C+2*B(3)*C^3-4*B(4)*C^4)/AZZ
1200 BZ(4)=(B(0)-B(1)*C+B(2)*C^2-B(3)*C^3+B(4)*C^4)/AZZ
1210 GOTO 1380
1220 REM-----
1230 REM   Fifth Order bilinear transformantion
1240 REM
1250 AZZ=B(0)+B(1)*C+B(2)*C^2+B(3)*C^3+B(4)*C^4+B(5)*C^5
1260 AZ(0)=(A(0)+A(1)*C+A(2)*C^2+A(3)*C^3+A(4)*C^4+A(5)*C^5)/AZZ
1270 AZ(1)=(5*A(0)+3*A(1)*C+A(2)*C^2-A(3)*C^3-3*A(4)*C^4-5*A(5)*C^5)/AZZ
1280 AZ(2)=(10*A(0)+2*A(1)*C-2*A(2)*C^2-2*A(3)*C^3+2*A(4)*C^4+10*A(5)*C^5)/AZZ
1290 AZ(3)=(10*A(0)-2*A(1)*C-2*A(2)*C^2+2*A(3)*C^3+2*A(4)*C^4-10*A(5)*C^5)/AZZ
1300 AZ(4)=(5*A(0)-3*A(1)*C+A(2)*C^2+A(3)*C^3-3*A(4)*C^4+5*A(5)*C^5)/AZZ
1310 AZ(5)=(A(0)-A(1)*C+A(2)*C^2-A(3)*C^3+A(4)*C^4-A(5)*C^5)/AZZ
1320 BZ(1)=(5*B(0)+3*B(1)*C+B(2)*C^2-B(3)*C^3-3*B(4)*C^4-5*B(5)*C^5)/AZZ
1330 BZ(2)=(10*B(0)+2*B(1)*C-2*B(2)*C^2-2*B(3)*C^3+2*B(4)*C^4+10*B(5)*C^5)/AZZ
1340 BZ(3)=(10*B(0)-2*B(1)*C-2*B(2)*C^2+2*B(3)*C^3+2*B(4)*C^4-10*B(5)*C^5)/AZZ
1350 BZ(4)=(5*B(0)-3*B(1)*C+B(2)*C^2+B(3)*C^3-3*B(4)*C^4+5*B(5)*C^5)/AZZ
1360 BZ(5)=(B(0)-B(1)*C+B(2)*C^2-B(3)*C^3+B(4)*C^4-B(5)*C^5)/AZZ
1370 REM-----
1380 REM Transformation complete
1390 GOTO 1680
1400 REM-----
1410 REM   Enter the digital transfer function
1420 CLS
1430 PRINT"DISCRETE-TIME TRANSFER FUNCTION"
1440 PRINT:PRINT
1450 INPUT"Input degree of Numerator polynomial";ND
1460 IF ABS(INT(ND))<>ND THEN 1440
1470 PRINT:PRINT
1480 PRINT"Numerator coefficients are assumed to be in the form:"
1490 PRINT
1500 PRINT"a(0)+a(1)*z^-1+a(2)*z^-2+...+a(k-1)*z^-(k-1)+a(k)*z^-k"
1510 PRINT:PRINT
1520 FOR I=0 TO ND
1530 PRINT"Input coefficient a(";I;")"
1540 INPUT AZ(I)
1550 NEXT I
1560 PRINT:PRINT
1570 INPUT"Input the degree of the Denominator Polynomial";DD
1580 IF ABS(INT(DD))<>DD THEN 1560
1590 PRINT:PRINT
1600 PRINT"Denominator coefficients are assumed to be in the form:"
1610 PRINT
1620 PRINT"1+b(1)*z^-1+b(2)*z^-2+...+b(k-1)*z^-(k-1)+b(k)*z^-k"
1630 PRINT:PRINT
1640 FOR I=1 TO DD
1650 PRINT"Input coefficient b(";I;")"
1660 INPUT BZ(I)
1670 NEXT I
1680 REM-----
1690 REM   Now we have the coefficients of the digital transfer
1700 REM   function. We must now send these coefficients to the micro-
1710 REM   controller inorder to dictate the respose of the motor.
1720 CLS
1730 PRINT:PRINT
1740 PRINT"The digital transfer function coefficients:"

```

```

1750 PRINT
1760 FOR I=0 TO 5
1770 PRINT"a( ";I;" ) =";AZ(I)
1780 NEXT I
1790 PRINT
1800 FOR I=1 TO 5
1810 PRINT"b( ";I;" ) =";BZ(I)
1820 NEXT I
1821 INPUT"enter the value of theta zero";THZ
1830 OPEN "coeff.h" FOR OUTPUT AS #1
1850 PRINT #1,"#define    a0    ";AZ(0)
1860 PRINT #1,"#define    a1    ";AZ(1)
1870 PRINT #1,"#define    a2    ";AZ(2)
1880 PRINT #1,"#define    a3    ";AZ(3)
1890 PRINT #1,"#define    a4    ";AZ(4)
1900 PRINT #1,"#define    a5    ";AZ(5)
1910 PRINT #1,"#define    b1    ";BZ(1)
1920 PRINT #1,"#define    b2    ";BZ(2)
1930 PRINT #1,"#define    b3    ";BZ(3)
1940 PRINT #1,"#define    b4    ";BZ(4)
1950 PRINT #1,"#define    b5    ";BZ(5)
1951 PRINT #1,"#define    thzero    ";THZ
1952 REM    add more define statements here inorder to raise the order
1953 REM    of the digital transferfunction.
1960 SYSTEM
ok

```

type dataio.h

```

/*****
 * dataio.h: This include file is designed to make it easier
 * to read and write analog data
 */

init_serial() /* This initializes the proper registers */
{
    /* for using the builtin serial port. */
    SCON = 0x52; /* Serial control register */
    TMOD = 0x20; /* Timer mode type register */
    TCON = 0x68; /* Timer control register */
    TH1 = 0xf3; /* Timer high byte register */
}

init_ad() /* This starts up the AD converter by */
{
    /* pulling /wr low for a time, and by */
    unsigned char _port1; /* grabbing and holding /cs low */
    P1 |= 0x0E;
    _port1 = P1 & 0xf1; /* /cs /wr /rd are high */
    P1 = _port1 | 0x04; /* pull /cs and /wr low */
    P1 = _port1 | 0x0c; /* let go of /wr */
}

unsigned char read_ad() /* This function reads a byte */
{
    /* from the internal latches of */
    unsigned char _port0, _port1; /* the AD converter. */
    _port1 = P1 & 0xf1;
    P0 = 0xff;
    P1 = _port1 | 0x08; /* Pull /cs and /rd low */
    _port0 = P0; /* Read port 0 */
    P1 = _port1 | 0x0c; /* let go of /rd */
    return(_port0);
}

write_da(_out) /* This function write a 10-bit */
unsigned int _out; /* integer to the DA converter */
{
    unsigned char _port1, _port2;
    _out &= 0x03FF;
    _port1 = ((_out & 0x0300) >> 4) | (P1 & 0xCF);
    _port2 = _out & 0xFF; /* We set up the MS bits of the integer on */
    P1 = _port1; /* bits 4&5 of port 1. We really need a 10-bit */
    P2 = _port2; /* latch in between the ports and the DA conv. */
}

```

A:\>

DOS C51 COMPILER V3.07, COMPILATION OF MODULE TFCNTRL
 OBJECT MODULE PLACED IN TFCNTRL.OBJ
 COMPILER INVOKED BY: C:\C51\BIN\C51.EXE TFCNTRL.C

stmt	level	source
1		#pragma SMALL
2		#pragma PRINT
3		#pragma CODE
4		
5		#include <reg5000.h> /* register definitions for 8051 CPU */
6		#include <stdlib.h> /* Has atof, and other functions */
7		#include <stdio.h> /* standard i/o definitions */
8		#include <dataio.h> /* a/d and d/a setup */
9		#include <coeff.h> /* control coefficients from trans4 */
10		
11		#define FALSE 0
12		#define TRUE 1
13		#define DELAY 60
14		
15		delay(ms) /* this routine is used to set up delay loops */
16		unsigned int ms;
17		{
18	1	unsigned int i, j;
19	1	for(i=0;i<ms;i++){
20	2	for(j=0;j<DELAY;j++);
21	2	}
22	1	}
23		
24		/****** User Inputed Transfer-Function Control *****/
25		/******
26		/* The controller is placed inside the while loop */
27		/* The a/d reads from 0 to 255 decimal */
28		/* The d/a writes from 0 to 920 decimal */
29		/******
30		/******
31		
32		main() {
33		
34	1	unsigned char port0;
35	1	unsigned int out,i;
36	1	float th,v,vo,aa,bb,y[10],u[10];
37	1	
38	1	P1 = 0x6e; /* P1 */
39	1	init_serial();
40	1	init_ad();
41	1	
42	1	putchar('0');
43	1	
44	1	
45	1	for (i=1;i<=6;i++){ /* initially set previous values to zero */
46	1	u[i]=0.;
47	2	y[i]=0.;
48	2	}
49	2	}
50	1	while (1) {
51	1	

```
52 2      delay(67);
53 2
54 2      port0=read_ad( );    /* read a value from the a/d */
55 2
```

```

56 2      th=((float)port0*.0246)-(float)thzero);
57 2
58 2      for (i=2;i<=6;i++)
59 2          u[i]=u[i-1];
60 2      u[1]=th;
61 2
62 2
63 2      /* IF HIGHER ORDER DIGITAL TRANSFER FUNCTIONS ARE REQUIRED */
64 2      /* SIMPLY ADD ON TO THE END OF aa AND bb */
65 2
66 2      aa=a0*u[1]+a1*u[2]+a2*u[3]+a3*u[4]+a4*u[5]+a5*u[6];
67 2      bb=b1*y[1]+b2*y[2]+b3*y[3]+b4*y[4]+b5*y[5];
68 2      v=aa-bb;
69 2
70 2      if (v < -10.)
71 2          vo=-10.;
72 2      else if (v > 10.)
73 2          vo=10.;
74 2      else
75 2          vo=v;
76 2
77 2      out=(int)(46.05*(10.-vo));
78 2
79 2      write_da(out); /* write an integer to the d/a */
80 2
81 2      for (i=2;i<=5;i++) /* update previous values */
82 2          y[i]=y[i-1];
83 2
84 2      y[1]=vo;
85 2
86 2  }
87 2  }
88 1  }
89 1  /*****

```

Appendix B
Digital Low Pass Filter Designs for Signal Processing

LOW PASS FILTER DESIGN FOR A CSI
TEST ARTICLE: NUMERICAL SIMULATION RESULTS

by

Nancy Nimmo
Spacecraft Dynamics Branch
NASA Langley Research Center
Hampton, VA 23665-5225

Gordon Lee
Mars Mission Research Center
North Carolina State University
Raleigh, NC 27695-7910

ABSTRACT

Multi-body flexible structures such as those currently under investigation in spacecraft designs, are large scale dimensional systems. As such, controlling and filtering issues for such structures are computationally complex in nature. One important issue of concern is the effects of filtering or pre-conditioning of the sensor data prior to application of the controller strategy. Phase distortion, resulting in nonlinear time delays, and spillover effects, due to finite order model and control approximations, may lead to undesirable results. This paper investigates several classical low pass filter designs under a new light, that is, with its effects on space structure control. Two control strategies - static dissipation control and virtual passive control - are investigated and the filter/controller configuration is applied to a numerical model of the control/structures Interaction Test Article - The Evolutionary Model - which is an experimental hardware currently being studied at NASA Langley Research Center.

I. Introduction

The next generation of spacecraft will be larger than current spacecraft and will require more sophisticated design, construction and operating methods. A sketch of a possible 21st century spacecraft is shown in Figure 1. Characteristics common to this class of spacecraft are long truss-type bodies with several appendages such as large solar arrays, radiators and circular

¹This research is supported in part by NASA Grant No. NAG-1331 to the Mars Mission Research Center.

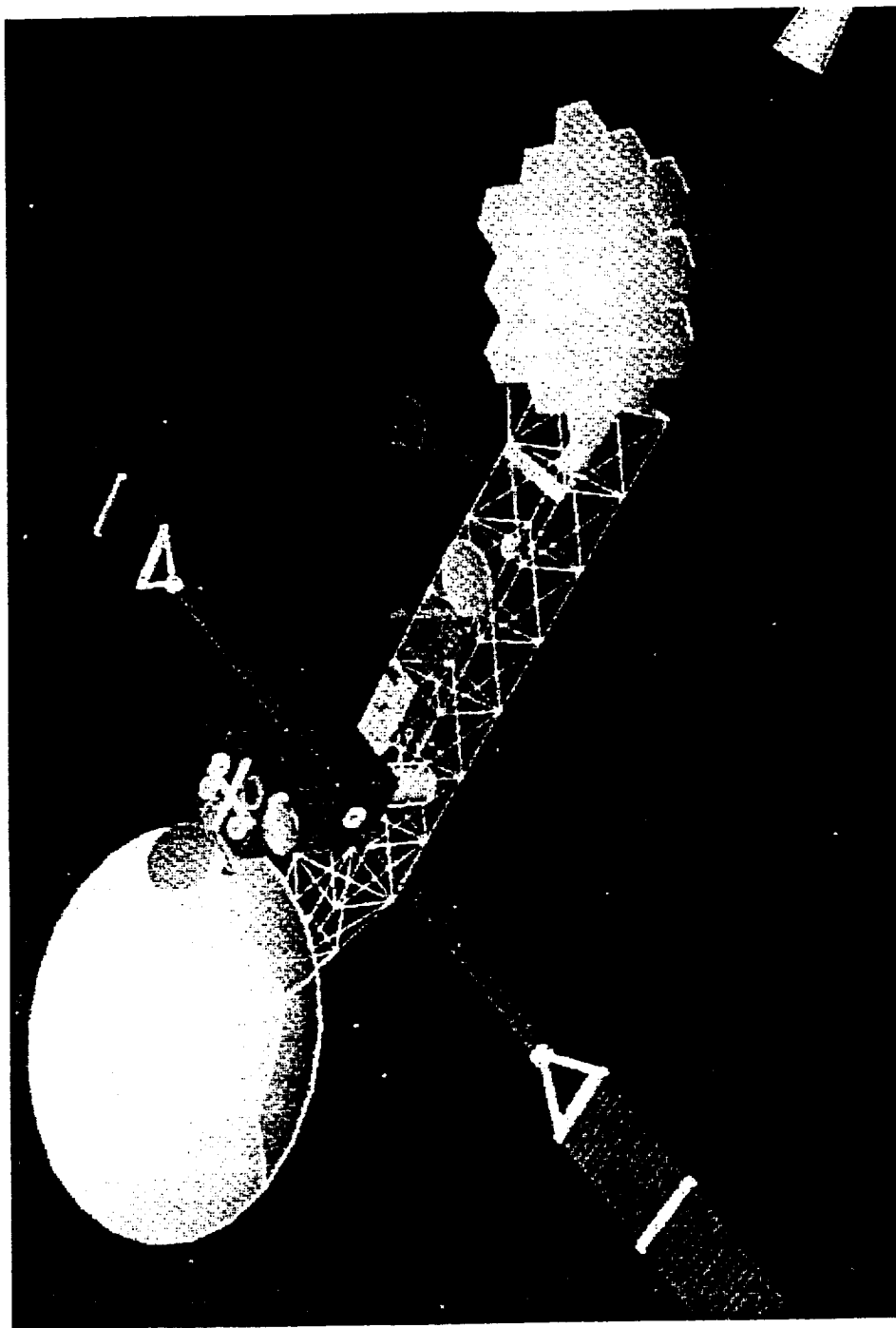


Figure 1: Mission to planet Earth geostationary platform

antennas. This type of spacecraft will probably contain a large number of lightly damped and closely spaced flexible modes at low frequencies. Traditional rigid-body control of spacecraft will not be adequate for these spacecraft. Rigid-body maneuvers such as attitude adjustment and slewing of appendages may excite flexible modes causing degraded performance and even instability or damage to the structure. This has motivated the study of active control methods for vibration suppression of large flexible space structures (LFSS).

The National Aeronautics and Space Administration (NASA) has developed the Controls Structure Interaction (CSI) program to address the problems associated with the design and control of LFSS. At the Langley Research Center's Spacecraft Dynamics Research Lab (SDRL), a testbed has been developed for system identification and closed-loop control experiments of CSI models.

One challenge regarding the control of LFSS is that accurate numerical models of large complicated structures are difficult to obtain. The controls designer often develops control laws which incorporate techniques for dealing with errors in the design model, and noise and disturbances in the input signal. Most of these techniques require some knowledge of the magnitude and bandwidth of these uncertainties. However, the complex, multibodied structures planned for future missions will make it increasingly difficult to predict these uncertainties. Hence it would be ideal to use filters to condition the sensor output so that control design would be less demanding. This is the motivation for a study of signal processing techniques, more specifically, low-pass sensor filters, for closed-loop control of LFSS.

The technology for conventional low-pass filters is well developed; they are well characterized, and easy to implement. With system identification and closed-loop control experiments, a first-order sensor filter is often used. In initial attempts to design a controller for the CSI Evolutionary model (CEM), a first-order low-pass Bessel filter was used to attenuate the response of a 34 Hz mode which was going unstable. The use of this filter allowed for a 40 percent increase in feedback gains before causing instability. Digital filtering was used in system

identification of another CSI test hardware, the Mini-Mast [1]. In both of these cases, the frequency above which attenuation is desired, or cutoff frequency, is set high so that there is little magnitude or phase distortion in the frequency range of interest.

Generally, higher-order filters are not used in closed-loop applications because the phase distortion negatively affects system stability. One example is an attempt to use multiple bandpass filters to isolate the modes to be controlled [2]. The frequencies of the modes to be controlled were close enough so that the rolloff from one bandpass filter affected the neighboring modes. This coupling between the filters and the modes produced system instability.

With the use of digital filtering, modifications to conventional filters can easily be made, and the use of higher-order filters may be possible. In research conducted on sensitivity of the Space Station Freedom's Alpha Joint controller to variations in the structural modal parameters, a modified fourth order Butterworth filter is used to attenuate the high frequency structural modes [3]. The modification consists of the addition of two zeros in the numerator of the filter transfer function to offset the phase distortion caused by the conventional filter. The filter dynamics is included within the controller optimization and the filter cutoff frequency is a design variable.

II. Controls Structures Interaction (CSI) Program Test Article

The test article selected as the reference problem for this paper is the CSI Evolutionary Model (CEM). The CEM testbed is well-supported and fully operational, thus providing an ideal environment for conducting research. In this section, a brief description of the CSI Program is first given, and then a description of the CEM test article and development of the analysis model follows.

The NASA CSI Program has been formed with the overall objective of developing and validating the technology needed to design, verify and operate spacecraft in which the structure and control system interact beneficially to meet the requirements of 21st-century spacecraft [4]. Traditional spacecraft control for maneuvering is designed for a considerably stiff structure. Problems may arise if this same type of control is attempted to maneuver large flexible space

structures (LFSS). Interaction between the structure and control system can reduce performance, restrict operations, and cause dynamic instabilities. The interaction between controls and structure may also be detrimental when the flexible appendages such as solar arrays or antennas are articulated. Hence the structure and control interactions must be considered in order to achieve desired system performance.

A CSI testbed has been developed in the Spacecraft Dynamics Research Lab (SDRL) at Langley Research Center. Some objectives of the testbed are to validate CSI design methodology, implement practical sensors and actuators for use in LFSS control, and to evaluate controller designs. This testbed, shown in Figure 2, consists of the CSI Evolutionary Model (CEM), suspension system, instrumentation for sensing, actuation, and data acquisition and processing capabilities. The CEM was designed to possess dynamic properties typical of space platforms proposed for remote sensing and communications. Line-of-sight (LOS) pointing is the primary objective for control system designs.

The CEM consists of a 52.5 ft. long main truss, a 16 ft. diameter reflector, and a 9.2 ft. long laser tower as shown in Figure 2. To monitor the LOS pointing accuracy, a low powered laser has been mounted on the laser tower such that the beam reflects upon a mirrored surface mounted on the reflector. A photo-diode array above the reflector measures the beam reflection to give LOS accuracy of the CEM [5]. There are 62 bays in the central truss, and the tower truss has 11 bays. Each bay is a 10 inch cube made of aluminum struts with single-laced diagonals. The reflector consists of 8 tapered ribs attached to a baseplate and bent to a preformed shape by a tension cable at their tip. The structure is suspended from the ceiling by two steel cables 64.5 ft. long. The total structure weighs 741 lbs.

For control and excitation purposes, eight actuators and eight accelerometers are placed in pairs at four locations along the truss (Figure 3). The actuators are compressed air thrusters which deliver proportional bi-directional force. Each actuator is capable of producing 2.2 lbs. of force. The control sensors are servo accelerometers with 5 volts/g. sensitivity.

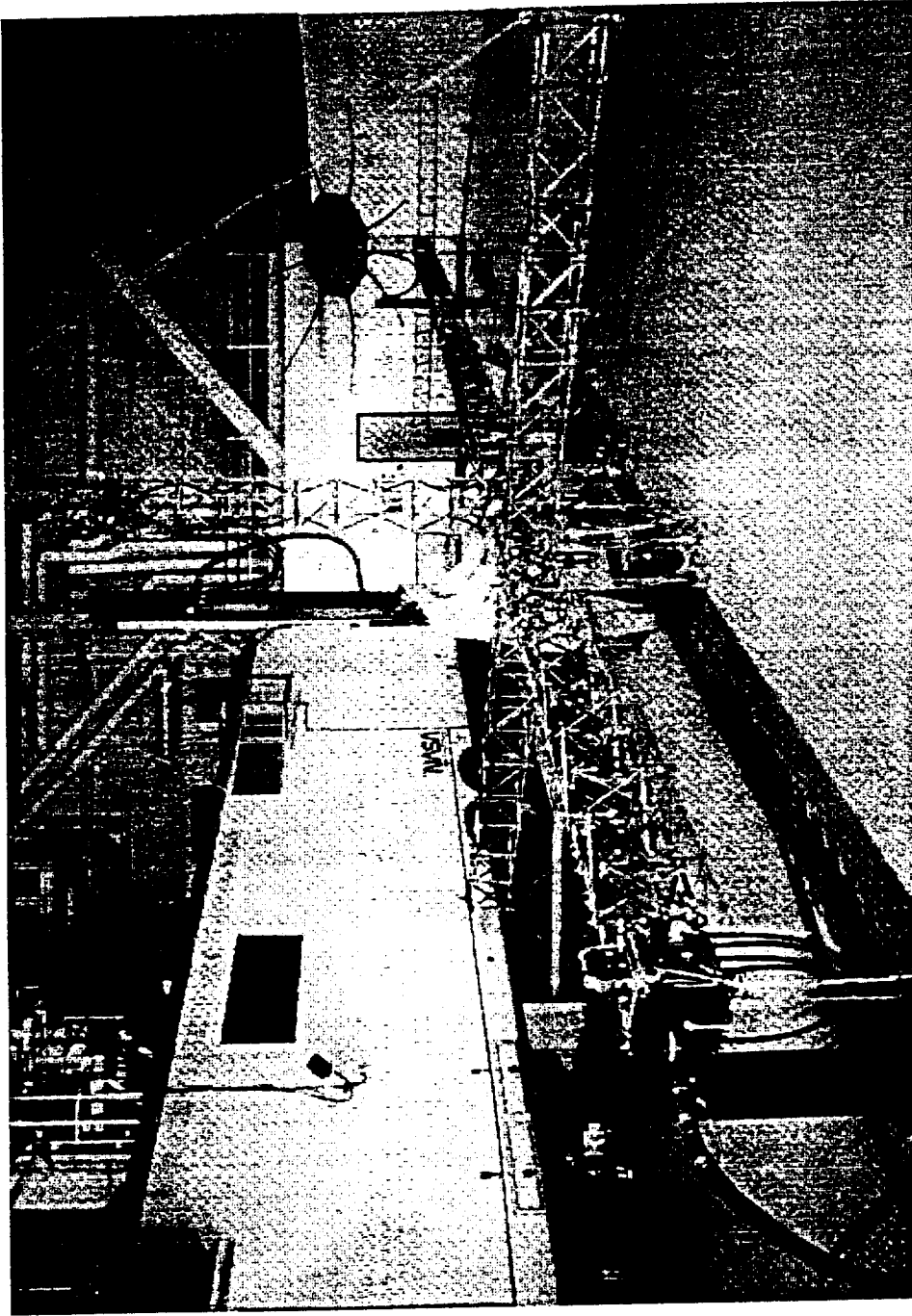


Figure 2: The CSI evolutionary model

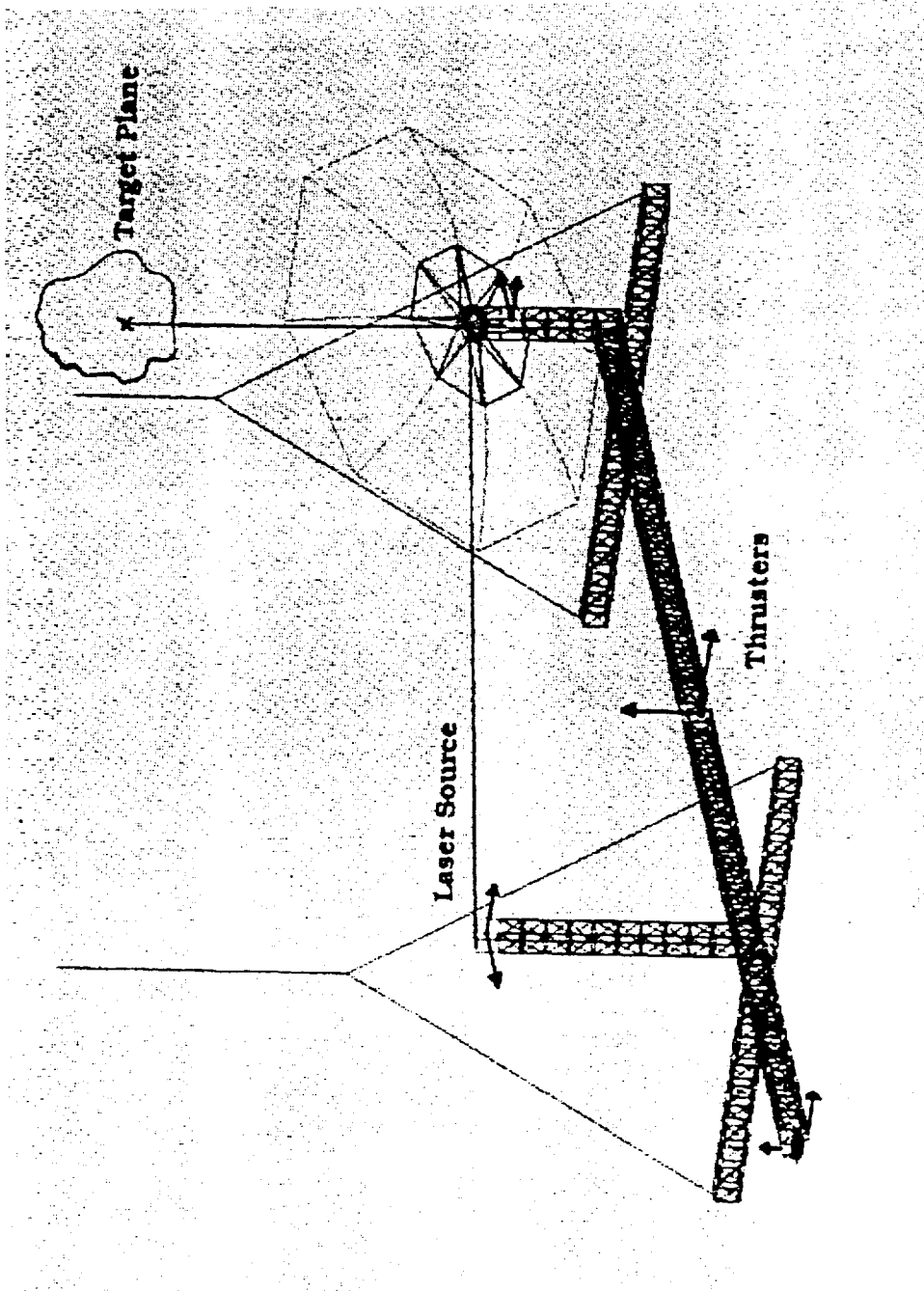


Figure 3: The CSI model showing actuator locations

The evaluation model used in this study is the most accurate mathematical model of the CEM currently available. For this reason it is also called *truth model*. This model is used to predict the behavior of the CEM for various scenarios such as response to a disturbance or response to various closed-loop control algorithms. There are several steps involved in the development of the evolution model, and each step involves the development of a different type of model. First, a detailed finite element model is developed. This is also referred to as an *analysis model*. Next, data from system identification tests used to develop an *experimental model*. Finally, information from the experimental model is used to improve the finite element model. For large complicated structures such as the CEM, this cycle may be repeated a number of times. Each time the cycle is completed, the analysis model becomes a more accurate mathematical representation of the physical model. For this study, the analysis model is used in *state-space form*, which is further described later in this section.

A. Finite element model

A detailed finite element model (FEM) of the CEM (Figure 3) was developed using the NASTRAN finite element analysis program. The FEM is made up of beam, rod, and plate elements with 609 grid points and contains over 3000 degrees of freedom. The suspension cables are included in the FEM and are modeled by rod and spring elements. Lumped masses are also included to represent the joints and actuators.

The analysis model predicts 81 modes of vibration below 50 Hz. Thirty of these modes are below 10 Hz [5]. Since the CEM is cable suspended, there are six rigid-body modes (also called pendulum modes) with frequencies between 0.1 Hz and 0.9 Hz. The first three flexible modes (modes 7, 8, and 9) involve bending and torsion of the CEM. These mode shapes are shown in Figure 4.

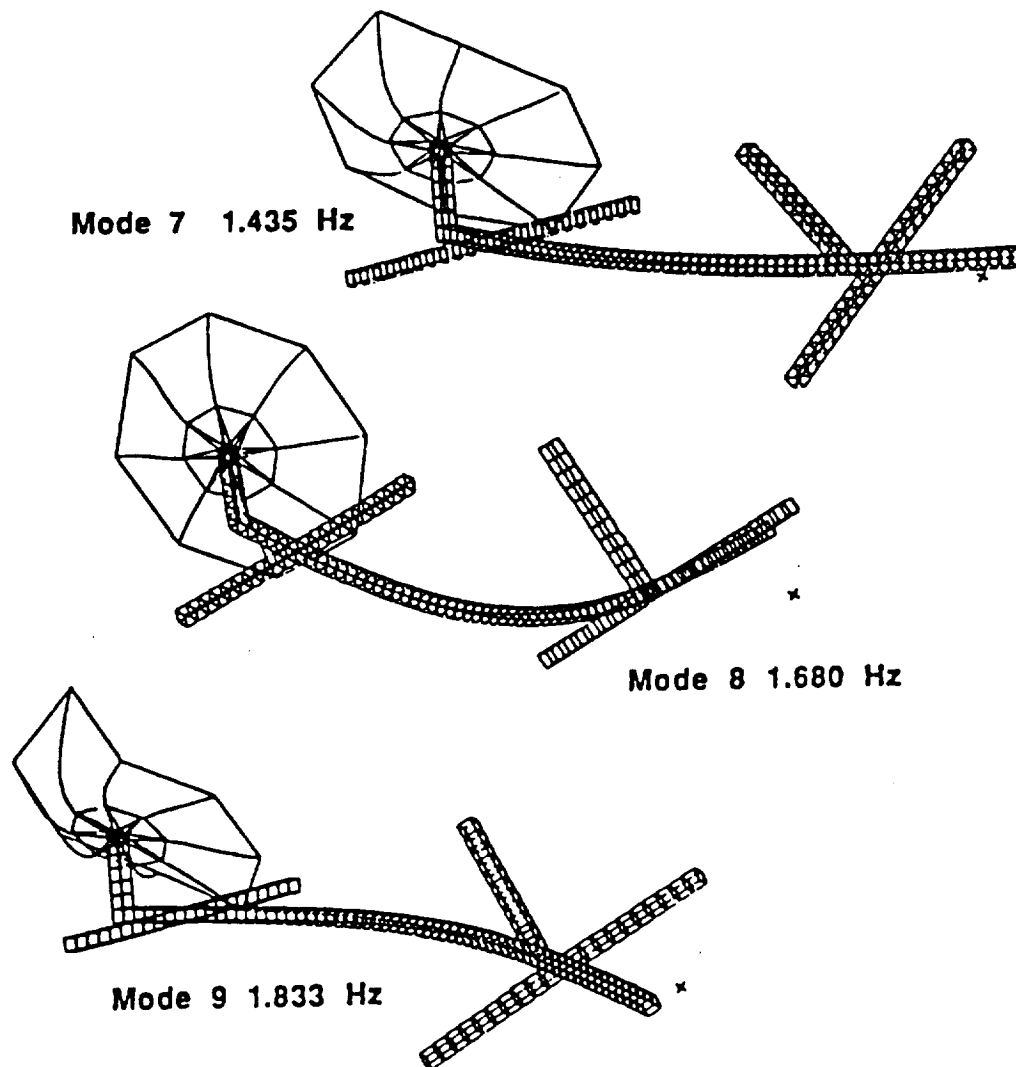


Figure 4: The first three flexible-body modes of vibration

B. Experimental Model

System identification tests are conducted to identify modal vibration parameters including frequencies, damping and mode shapes. These identified parameters define the experimental model. System identification for the CEM consists of multi-input, multi-output (MIMO) tests, and extraction of the frequency response functions (FRF) between the acceleration output to force input. Simply defined, an experimental FRF is found by dividing the Fast Fourier Transform (FFT) of the output by the FFT of the input. Typical FRF's from the center of the main truss, in the horizontal and vertical planes, are shown in Figure 5 [5]. Also shown on these plots is the

some of the dominant modes; however refinement of the FEM is needed. The modal vibration parameters for nine modes have been identified at the present time. These parameters were used to refine the FEM and is included in the current evaluation model. Further system identification tests are underway to better identify the modal parameters for flexible modes below 10 Hz.

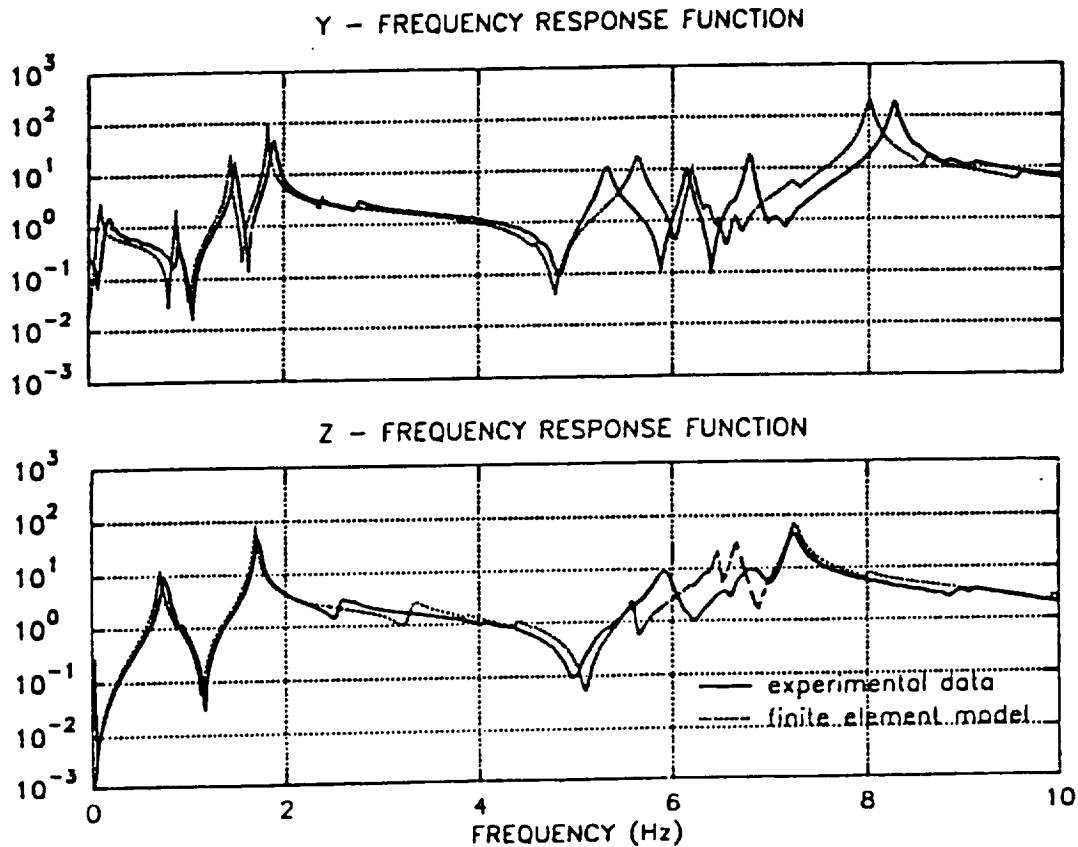


Figure 5: Frequency response functions for CEM

C. State-space model

The state-space form of the analysis model contains the first 40 modes from the refined FEM. The process of transforming the FEM to state-space form is now described.

An FEM with n physical degrees-of-freedom (dof), m actuators (inputs) and m sensors (outputs) can be represented by the following second order dynamic system:

$$M\ddot{z} + C\dot{z} + Kz = Qu \quad \text{and} \quad y = P\dot{z} \quad (1)$$

where M is an $n \times n$ mass matrix, C is an $n \times n$ damping coefficient matrix, K is an $n \times n$ stiffness matrix, and z is an $n \times 1$ vector containing the physical dof of the model. Further, u is an $m \times 1$ input vector, Q is the configuration matrix for the actuator locations, y is an $m \times 1$ output vector (in terms of acceleration since the output sensors are accelerometers), and P is the configuration matrix for the sensor locations.

Note n system eigenvalues λ , and n corresponding eigenvectors v , are calculated using finite element analysis. The number of modes to be kept for the state-space model is l and is less than n . Thus the modal matrix V has as its columns l $n \times l$ eigenvectors, and is orthogonal with respect to M and K . Hence V can be normalized so as to satisfy

$$V^T M V = I, \quad \text{and} \quad V^T K V = \Lambda \quad (2)$$

where $\Lambda = \text{diag}[\lambda_i] = \text{diag}[\omega_i^2]$. ω_i ($i = 1, 2, \dots, l$) are natural frequencies of the system. If a vector is defined as $z = V\eta$, where η is the $l \times 1$ modal coordinate vector, then substitution by the modal vector into Eq. (1) and premultiplication by V^T results in the modal system

$$\ddot{\eta} + C'\dot{\eta} + \Lambda\eta = Q'u \quad (3a)$$

$$y = PV\ddot{\eta} = -PV(C'\dot{\eta} + \Lambda\eta) + PVQ'u \quad (3b)$$

where $C' = V^T C V$ is a real symmetric $l \times l$ matrix, and $Q' = V^T Q$ is a real $l \times m$ matrix. The system is now in reduced modal coordinates.

If an $l \times l$ state vector x is defined as $x = [\eta^T \quad \dot{\eta}^T]^T$, Eq. (1) can be rewritten in the following state-space form:

$$\dot{x} = Ax + Bu \quad \text{and} \quad y = Cx + Du \quad (4)$$

where

$$A = \begin{bmatrix} 0 & I \\ -\Lambda & -C' \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ Q' \end{bmatrix} \quad (5a)$$

$$C = [-PVA \quad -PVC'] \quad \text{and} \quad D = PVQ' \quad (5b)$$

The state matrices A , B , C , and D are $2l \times 2l$, $2l \times m$, $m \times 2l$, and $m \times m$, respectively. The state-space form of the analysis model is used extensively in this study. A further treatment of the transformation from a FEM in physical coordinates to a state-space modal model is found in [6]. The actual values of (A, B, C, D) for the CEM are given in [7].

III. Selection of Filter Specifications

Before a filter is designed, a set of critical frequencies and the attenuation desired at these frequencies must be specified. These filter specifications determine the filter order, and in some cases, the cutoff frequency (such as with Butterworth and Bessel filters). The selection of these specifications depends on the frequency characteristics of the signal to be filtered, and the application of the filtered signal. For the reference problem, low-pass filtering of accelerometer signals to be used in closed-loop control applications is desired.

Frequency response functions (FRFs) from system identification tests are examined for frequency content of CEM accelerometer outputs. A large number of significant spikes appear in the 0 to 10 Hz range. Some of these represent the pendulum and structural modes to be controlled. There are a number of significant spikes, such as one at 34 Hz, as well as many smaller spikes above 10 Hz. Some of these are structural modes, and many are attributed to high frequency noise in the accelerometers.

In preliminary closed-loop control tests, a 34 Hz mode became unstable in the attempt to control nine modes with frequencies below 2 Hz [10]. A possible solution for this problem is to use a low-pass filter to attenuate this 34 Hz frequency component. This suggests a stopband frequency of 30 Hz. A significant portion of the frequency components above 30 Hz needs to be rejected so a 90% loss (20 dB) is specified as the minimum attenuation desired in the stopband.

For this problem, the passband frequency and maximum attenuation are more difficult to specify. Ideally, all frequency components above the frequency of the highest controlled mode (1.9 Hz for the CEM) would be rejected. Thus, the passband frequency for an ideal filter is 2 Hz. However, since practical filters exhibit rolloff near the passband frequency, the passband

frequency needs to be significantly higher than 2 Hz. Also, the phase loss near the cutoff frequency could significantly distort the frequency components of the modes to be controlled. If the passband frequency is specified as 15 Hz, there is minimal phase loss in the region below 2 Hz; also the phase response of the filter in this region is fairly linear. Maximum attenuation in the passband is specified as 0.5 dB since accuracy of the frequency components of the controlled modes is desired. For the Butterworth and Bessel filters, the cutoff frequency is specified as 15 Hz. Although there are 3 dB of attenuation at the cutoff frequency for these two filters, with the cutoff frequency set at 15 Hz, attenuation in the region below 2 Hz is less than the specified 0.5 dB. Now the filter specifications can be designated in the following manner:

$$f_p = 15 \text{ Hz}, \quad \alpha_{\max} = 0.5 \text{ dB}, \quad f_s = 30 \text{ Hz}, \quad \alpha_{\min} = 20 \text{ dB}$$

and are illustrated in Figure 6. The dashed curve represents the frequency response of a low-pass filter that meets the specifications. For the numerical simulations, Butterworth, Chebyshev,

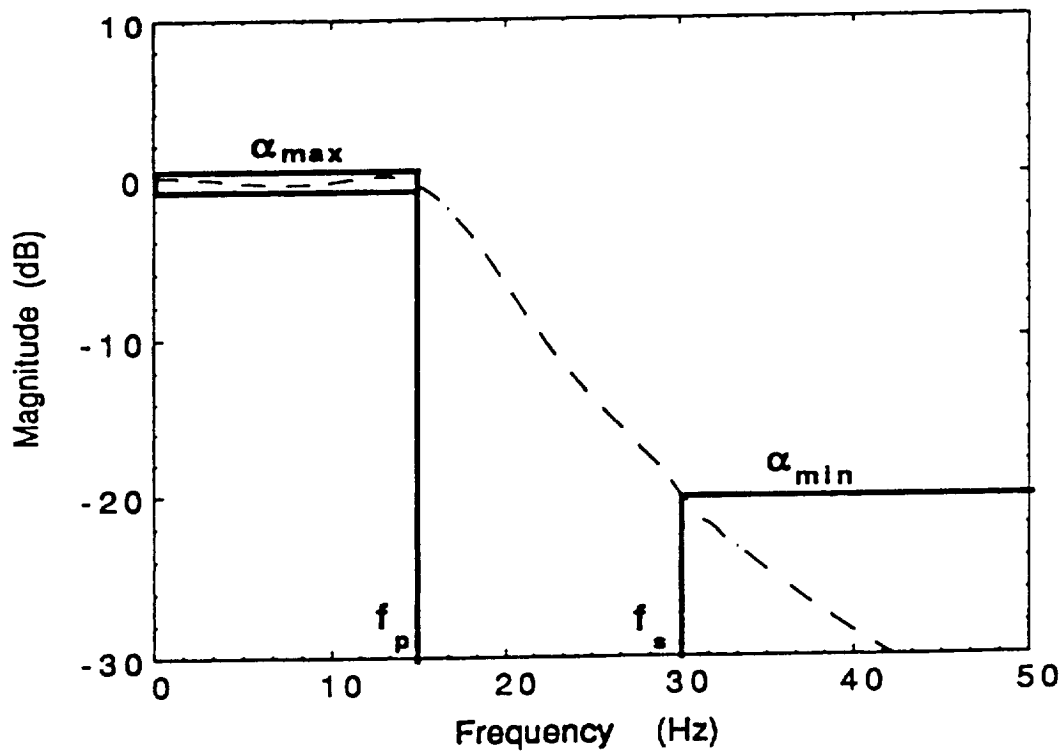


Figure 6: Filter specifications for CSI evolutionary model

Cauer, and Bessel filters are designed according to these specifications, and used to filter the simulated acceleration output from the CEM.

IV. Open-Loop Simulation Results

The numerical open-loop tests (system plus filter) are conducted to extract the frequency response of the system when subject to an impulse force. The frequency response functions for the open-loop system are calculated for fifteen CEM and filter combinations. The first step is to compute the frequency response of the system without any sensor filters. This response is compared to frequency responses of all the other system configurations. The other system configurations include the CEM with either a low-pass Butterworth, Chebyshev, Cauer, or Bessel filter. The frequency range examined is between 0 and 50 Hz. The resulting complex responses are plotted in phase and magnitude form.

The first set of frequency response functions show the effect of using second, fourth and sixth-order Butterworth filters with the CEM. The magnitude and phase of these frequency responses, and that of the unfiltered system is plotted in Figure 7. The magnitude responses in the region below 10 Hz are indistinguishable from each other. The frequency requirements below 2 Hz are satisfied by each filter. The magnitude response of the systems with fourth and sixth-order Butterworth filters fulfill the requirement that the magnitude of the frequency components above 30 Hz be attenuated by 20 dB (90% loss). Thus a fourth-order Butterworth filter is needed to satisfy the filter specifications. The phase loss caused by the Butterworth filters increases significantly with filter order. In the region below 2 Hz the loss appears to be linear.

With a system with many modes it is somewhat difficult to specify relative stability of a system using the phase and gain margins. However, trends in the frequency responses of the systems with various filters can be identified. There is good relative stability of the system without a filter, and even with the second-order Butterworth filter. This is illustrated by examination of Figure 7. The phase response in Figure 7b for the system without a filter, and with the second-order filter never exceeds 180 degrees in the frequency range shown. Thus there is some positive

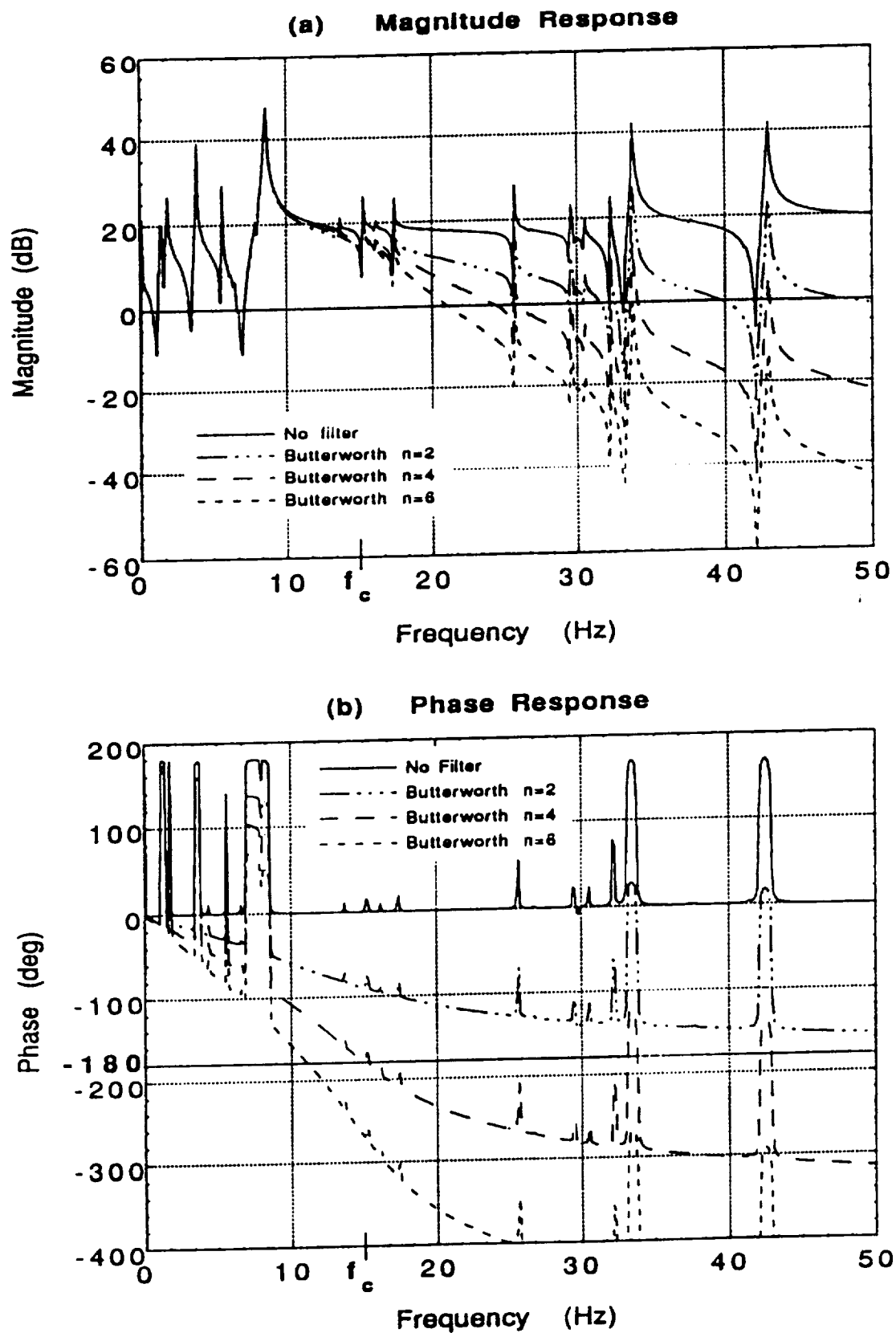


Figure 7: Frequency response of CSI evolutionary model with Butterworth filters

phase margin at any gain crossover frequency, and the gain margin is infinite. However, one can see that the addition of any filter, especially the higher-order filters, causes the phase margin to decrease significantly. The phase crossover frequency of the system with the fourth-order Butterworth filter is approximately 16 Hz, and that of the sixth-order filter decreases to approximately 12 Hz. The general trend is that the relative stability deteriorates with increased filter order.

The second set of frequency response functions show the effect of using second, fourth and sixth-order Chebyshev filters with the CEM. The magnitude and phase of these frequency responses, and that of the unfiltered system is plotted in Figure 8. A third-order Chebyshev filter satisfies the specified requirements of at least 20 dB loss in the stopband, and a maximum attenuation of 3 dB in the passband. When the maximum attenuation is reduced to 0.5 dB, the slope of the magnitude response decreases in the transition band. Thus for a third-order Chebyshev filter with 0.5 dB maximum attenuation in the passband, the attenuation at the stopband frequency (30 Hz) is less than 20 dB (approximately 19 dB). To satisfy these specifications, a fourth-order Chebyshev filter is needed. The fourth-order Chebyshev filter meets the required specifications as seen in Fig. 8a.

The trend of deteriorating relative stability of the responses of the systems with the fourth and sixth-order Chebyshev is similar to that of the Butterworth filters. The phase crossover frequency for the fourth-order Chebyshev system is approximately 14 Hz, and for the sixth-order system, 9 Hz.

Figure 9 shows the frequency responses for the CEM with second, fourth and sixth-order Cauer filters. The second-order Cauer filter satisfies the specifications. As with the Chebyshev filter, the decrease in maximum attenuation in the passband from 3 dB to 0.5 dB causes the slope of the phase response to decrease. A third-order Cauer filter is required to satisfy the specification that a minimum of 20 dB attenuation be attained by the stopband frequency (30 Hz). Although the frequency response of the CEM with a third-order Cauer filter is not shown, it is obvious that at 30

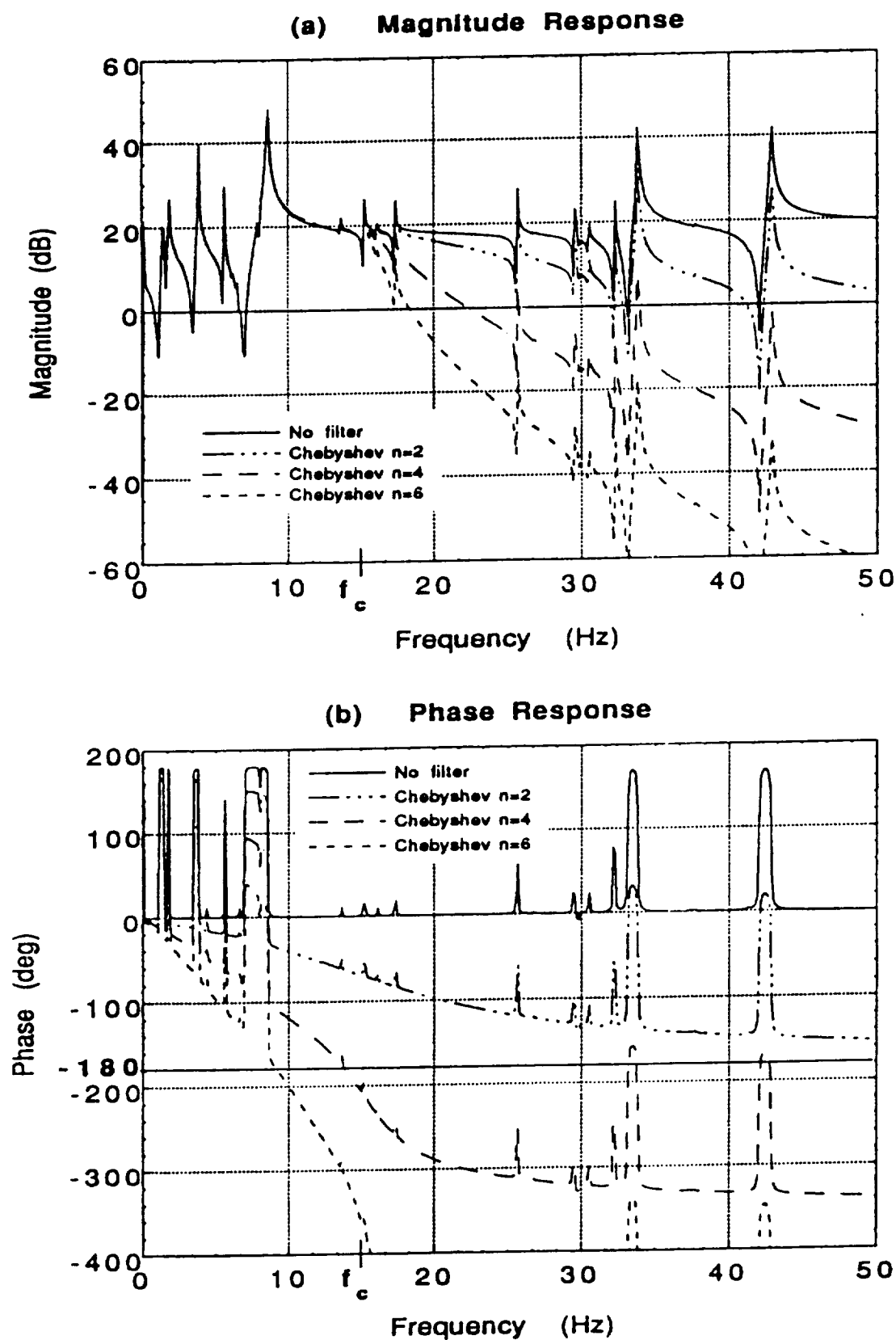


Figure 8: Frequency response of CSI evolutionary model with Chebyshev filters

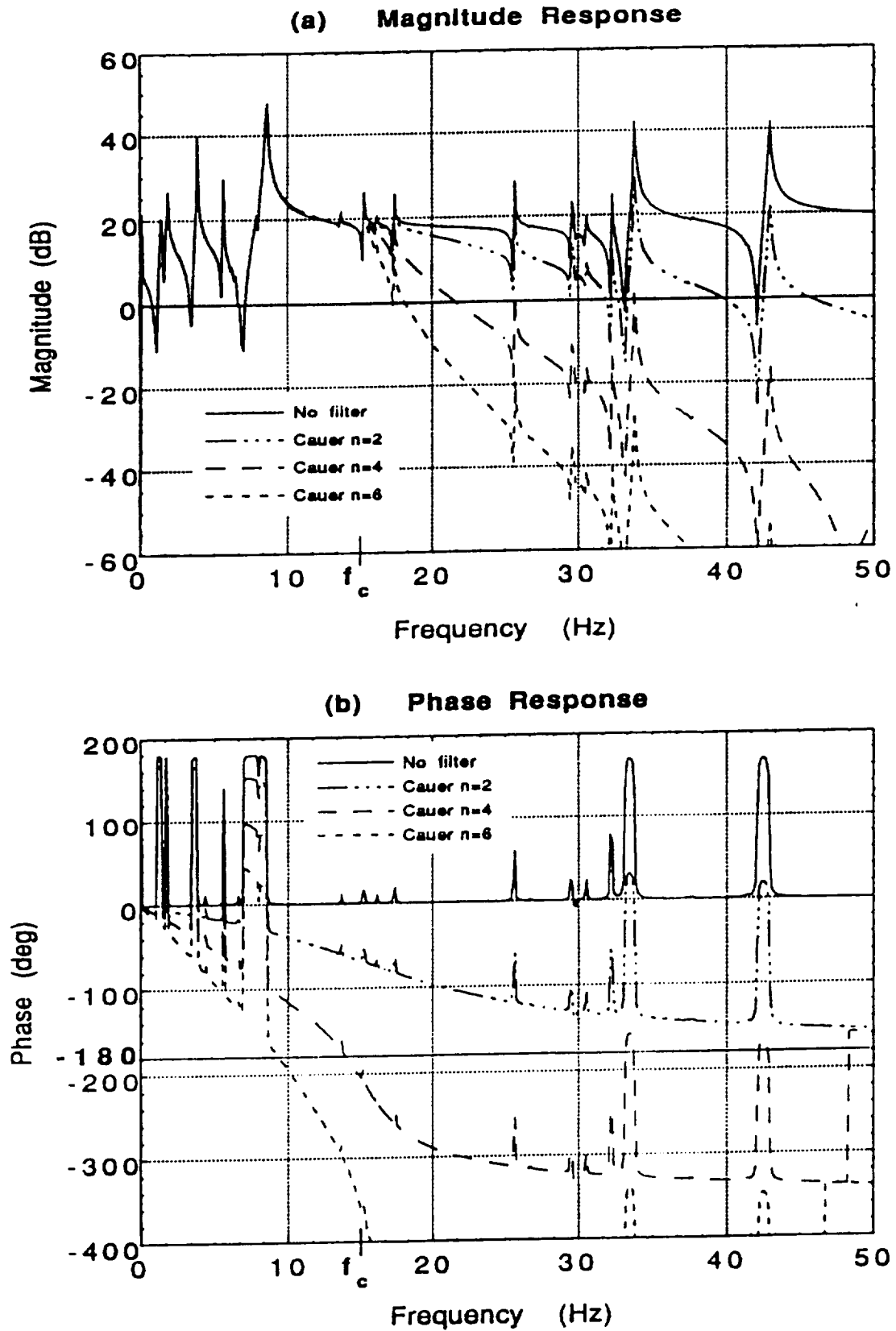


Figure 9: Frequency response of CSI evolutionary model with Cauer filters

Hz, the response of the system with a second-order filter is not 20 dB below that of the system without a filter.

The frequency response of the systems with the Cauer filters are very similar to that of the Chebyshev systems. However, the attenuation rate in the transition band of the Cauer systems are slightly higher than with the Chebyshev system. Also, due to the 'ripple' in the stopband of the Cauer filter, the magnitude response in the stopband is slightly more distorted. The same observations given about relative distortion of the Chebyshev systems can be given for the Cauer systems. The phase crossover frequency for the system with the four and sixth-order Cauer filter is approximately 14 Hz and 9 Hz, respectively.

The last set of frequency response functions are for open-loop systems consisting of the CEM and second, fourth and sixth-order Bessel filters. The magnitude and phase responses for these systems are shown in Figure 10. A tenth-order Bessel filter satisfies the specification of 3 dB attenuation at the passband frequency but does not attain 20 dB loss at the stopband frequency. As with the Butterworth filters, in the range below 2 Hz, the Bessel filters cause less than 0.5 dB attenuation of the CEM output. The sixth-order filter provides approximately 17 dB of attenuation (86% loss) at the stopband frequency. Since higher order Bessel filters do not provide much additional attenuation at the stopband frequency, if 20 dB loss at 30 Hz is critical, then a Bessel filter is not appropriate.

As with the other open-loop systems including filters, relative stability deteriorates with the addition of Bessel filters to the system. The phase crossover frequency for the fourth and sixth-order filters are approximately 22 Hz and 18 Hz, respectively. The Bessel filter causes the least amount of phase loss, and has the most linear phase response of all the filters considered.

The difference in the frequency content of a filtered and unfiltered accelerometer signal from actual laboratory tests is shown in Figure 11. In the two tests, several dominant modes of the CEM were excited and then allowed to decay freely. In the first test, sensor filters were not used. In the second test, second-order Cauer filters were used to filter accelerometer output. FFT's were performed on the free decay portion of the output. The magnitude of the frequency response is

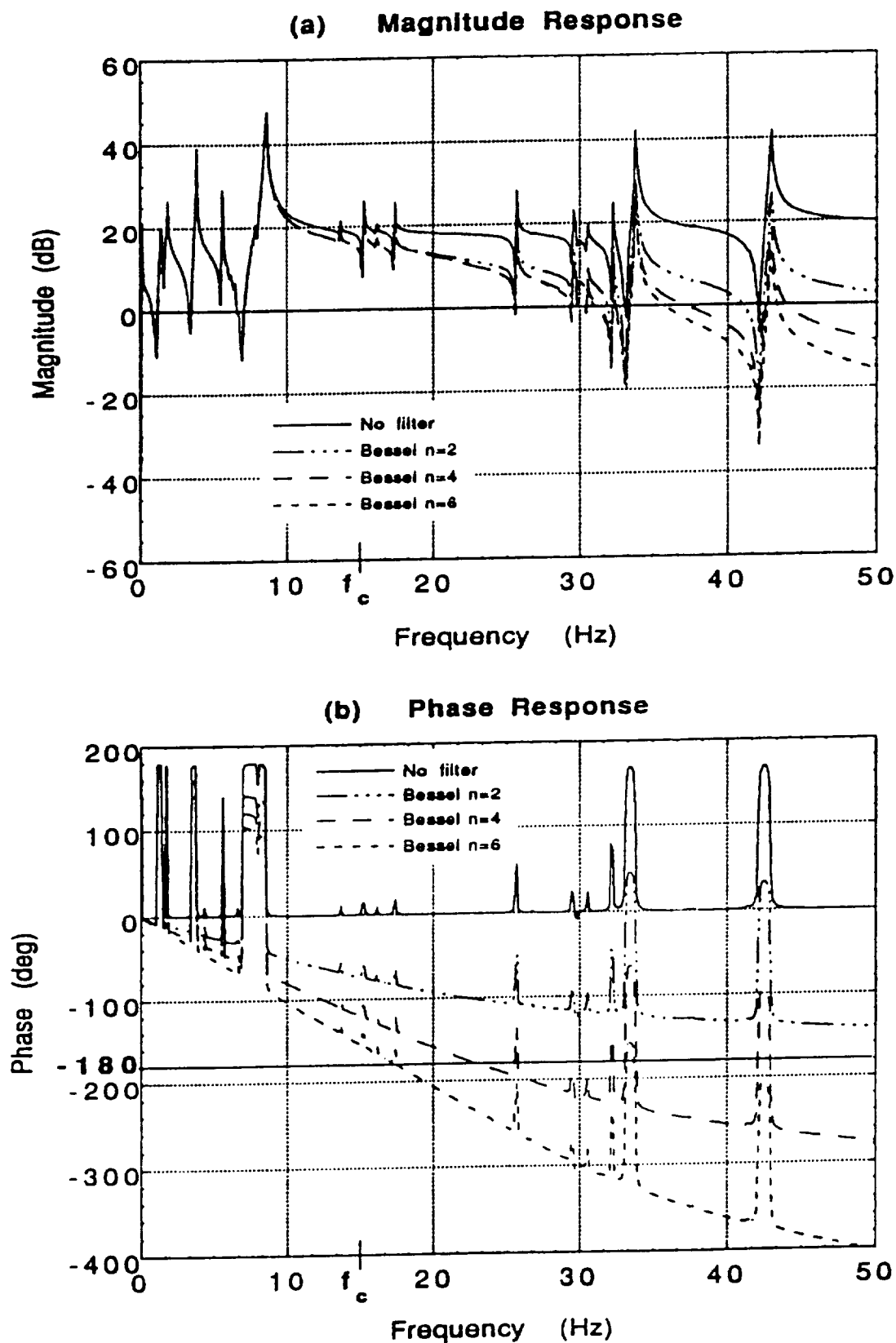
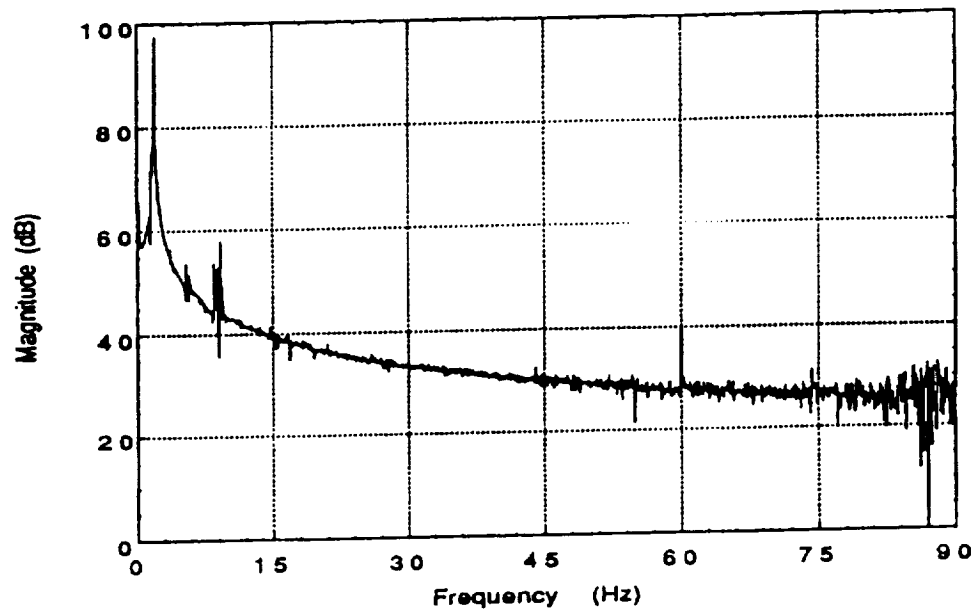
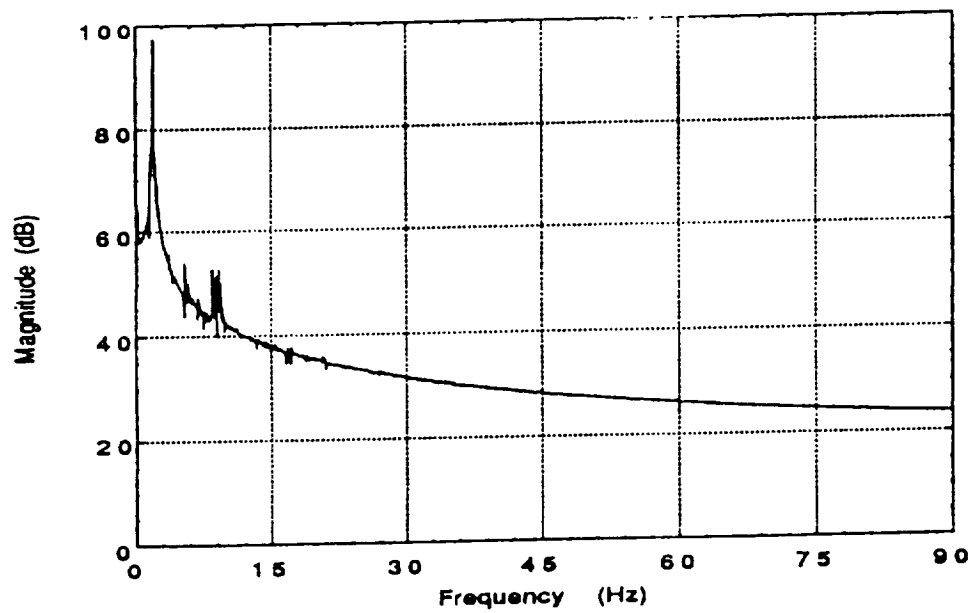


Figure 10: Frequency response of CSI evolutionary Model with Bessel filters



(a) Unfiltered output



(b) Output filtered with second-order Butterworth filter

Figure 11: FFT's of accelerometer signals from open-loop laboratory tests

found by calculating the absolute value of each FFT function. There is a good amount of activity in the upper half of the magnitude response in Figure 11a. This can be attributed to noise, disturbances and other unknown factors. In Figure 11b, the portion of the magnitude response above the cutoff frequency (15 Hz) is much smoother compared to that of Figure 11a. The Causer filter has successfully filtered out the unwanted frequency components of the signal.

By examining the frequency responses of the filtered and unfiltered open-loop systems, the effect of a sensor filter on the phase and magnitude of the CEM output can be seen. As higher-order filters are added to the system, the relative stability of the system decreases. This will greatly affect the ability to use the system in closed-loop control applications.

V. Control Strategy

Two objectives for control of large flexible space structures (LFSS) are accurate line-of-sight pointing and vibration suppression. Classical control designs depend on the accurate prediction of the dynamics of the structure. However, with large multi-bodied, complex structures, accurate models are difficult to develop. The models of LFSS available for control design usually contain inaccurate model parameters (such as modal damping coefficients). The problem of unmodeled high-frequency flexible modes is another challenge of control design. Although a detailed model for the CEM has been developed and preliminary identification has increased the accuracy of the first nine modes, there is still enough model error so that classical control methods are ineffective, or even worse, cause system instabilities. For these reasons, robust control designs are used.

The two control designs used in this study utilize the knowledge of the *design model* to obtain maximum performance, but do not depend on model accuracy to maintain stability. The design model is the model which is used to design the controller. The first method, static dissipative control [8], is a local velocity feedback method, and has been implemented by the CSI Analysis Methods Group. The second method, virtual passive control [9], is a second-order controller using direct acceleration feedback and has been implemented by the CSI Ground Test Group [10].

VI. Closed-Loop Simulation

Sensor filters have been successfully used in the open-loop experiments discussed in Section IV. By examining the FFT curves in Figure 11 one can see how effectively a second-order Butterworth filter can condition the accelerometer signal. However, when closed-loop control of a LFSS is desired, filtering a signal can have a significant effect on the stability of the system. Due to this reason, experiments involving control of LFSS are usually conducted without extensive prefiltering of the sensor signal. Some systems may include a first-order filter; however, a first-order filter is not very effective. With the larger, more complex, multibody space structures envisioned for the future, it may be necessary to include sensor filters so that closed-loop control of the structure can be accomplished more effectively.

Numerical and laboratory experiments were conducted to investigate the effect of sensor filtering on performance of feedback controllers. All four low-pass filters, Butterworth, Chebyshev, Cauer, and Bessel have been initially investigated while the Butterworth and Cauer filters were used in further investigations. The Butterworth filter is used because it is a popular, easily implemented filter, incurring relatively conservative phase distortion. The Cauer filter is the other low-pass filter selected for study because it provides the best magnitude response characteristics; therefore it would be ideal to use for our purposes if the resulting phase distortion could be managed.

Two types of numerical analysis are conducted on the closed-loop systems. The first type of analysis consists of calculating the eigenvalues of the system. The second type of analysis consists of calculating the response of the system to a given excitation and examining the time histories of these responses. Presented in this section are the results of numerical analysis of the closed-loop systems. The results of laboratory experiments are discussed in Section VII.

The experiment to be simulated, consists of excitation of the structure for seven seconds, free decay for three seconds, and then closed-loop control is initiated to damp out vibrations of the system. Discretized simulations are conducted in an attempt to produce results that will match that of laboratory experiments. A sampling rate of 250 Hz is used because this provides a folding

frequency of 125 Hz which is well above the frequency of the highest predicted structural mode (approximately 46 Hz).

The excitation consists of sinusoidal force input at four locations, one at 1.7 Hz (bending mode), one at 1.9 Hz (bending/torsion mode) and two at 0.145 Hz (first pendulum mode). Control force is applied at the eight thruster locations. Acceleration output is calculated for all eight accelerometer locations. Simulated acceleration at sensor location 1 (Figure 3) is shown in all the time history plots.

The plots show acceleration output using the Static Dissipative Controller (SD) and Virtual Passive Controller (AVA) for various filters. The notation in the captions denote the filter type and filter order.

In Figure 12b, controller SDB4 is used to control a system with fourth-order Butterworth filters. The response from each closed-loop system is stable, including that the response from systems with the sixth-order filters (Figures 12c and 13c). However, the performance of the controllers in the systems with the higher-order filters are obviously reduced. This is especially true for the static dissipative controllers developed using the sixth-order Cauer filter (SDC6), and sixth-order Butterworth filter (SDB6).

In Figures 15 and 16, the output of the system using controller AVA, and second and third-order Butterworth and Cauer filters are plotted. The unstable modes calculated with eigenvalue analysis (12.4 Hz, and 16.6 Hz) are only slightly noticeable in the systems with third-order filters. The eigenvalue representing an unstable mode is very close to the imaginary axis, so this mode could be considered only marginally unstable.

In summary, all closed-loop cases illustrate that the filters perform this appropriate function and provide to the controller a clear signal to be controlled. The closed-loop program is stable and whatever phase distortion that is introduced by the filters have minimal effect on the controller's performance. This may not be true, in general, however, and an all-pass filter may be required to remove the nonlinear phase characteristics introduced by the filters.

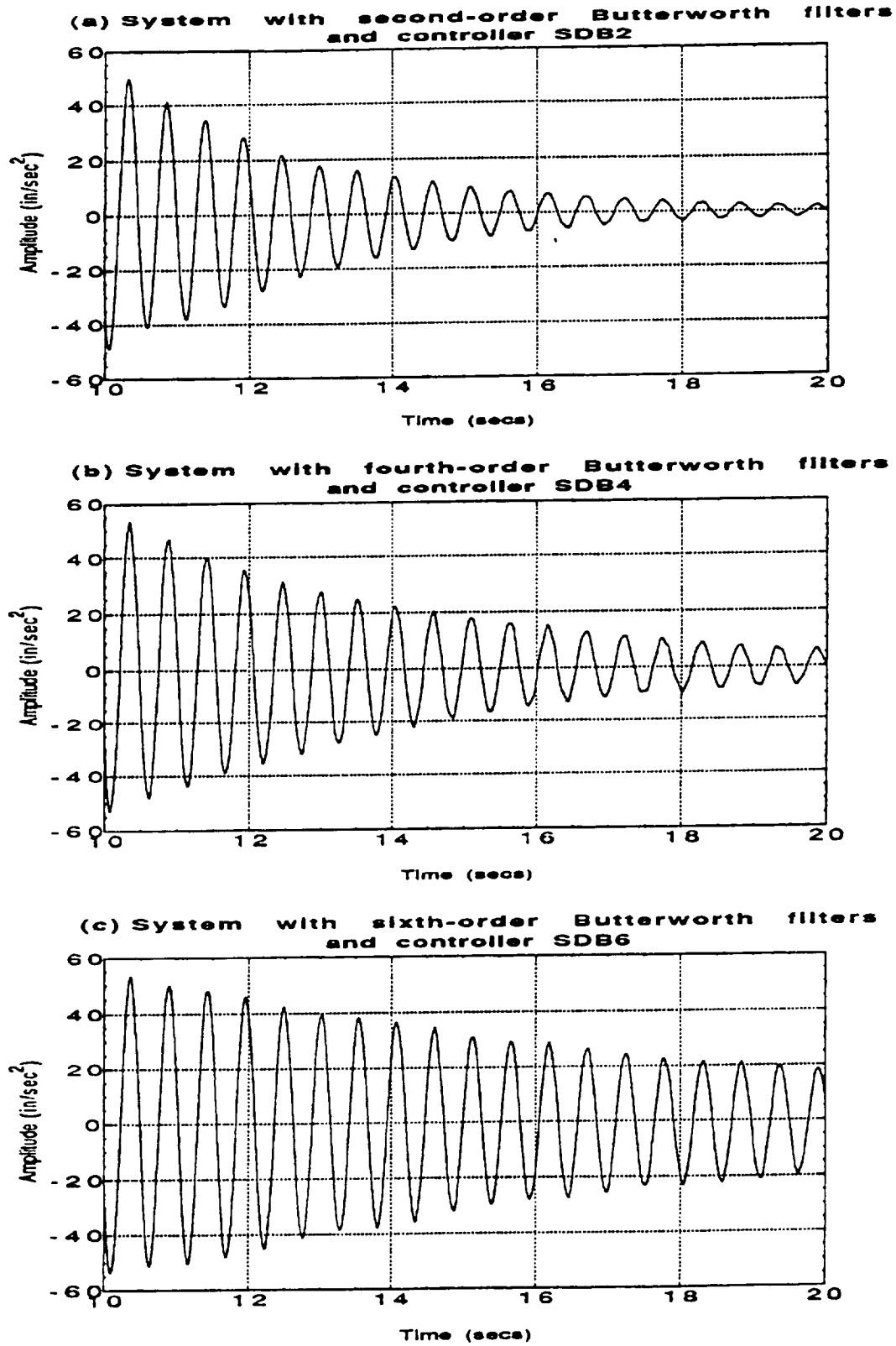


Figure 12: Simulated acceleration response of closed-loop systems with controllers developed with filter dynamics in the design model (sensor location 1)

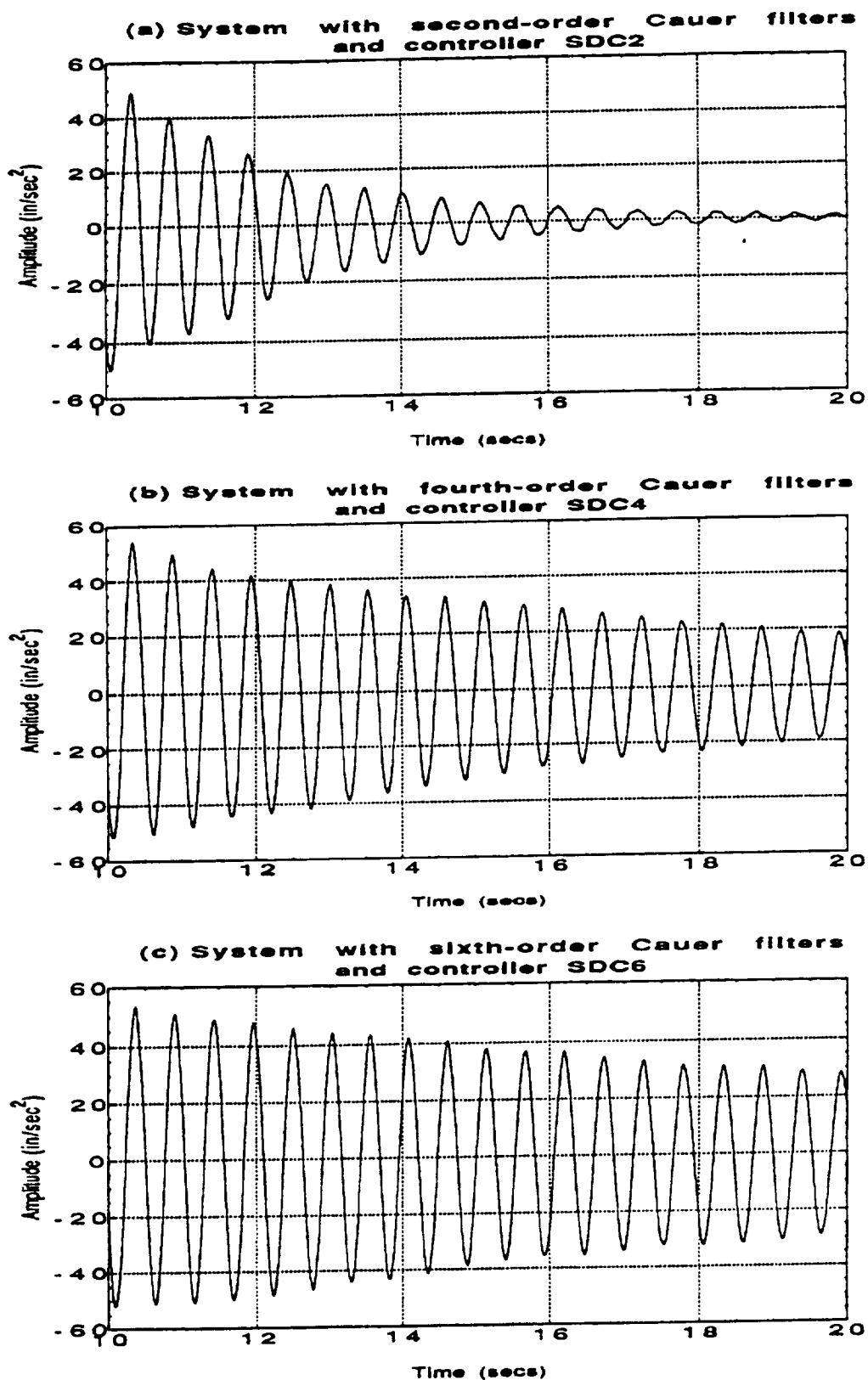


Figure 13: Simulated acceleration response of closed-loop systems with controllers developed with Cauer filter dynamics in the design model (sensor location 1)

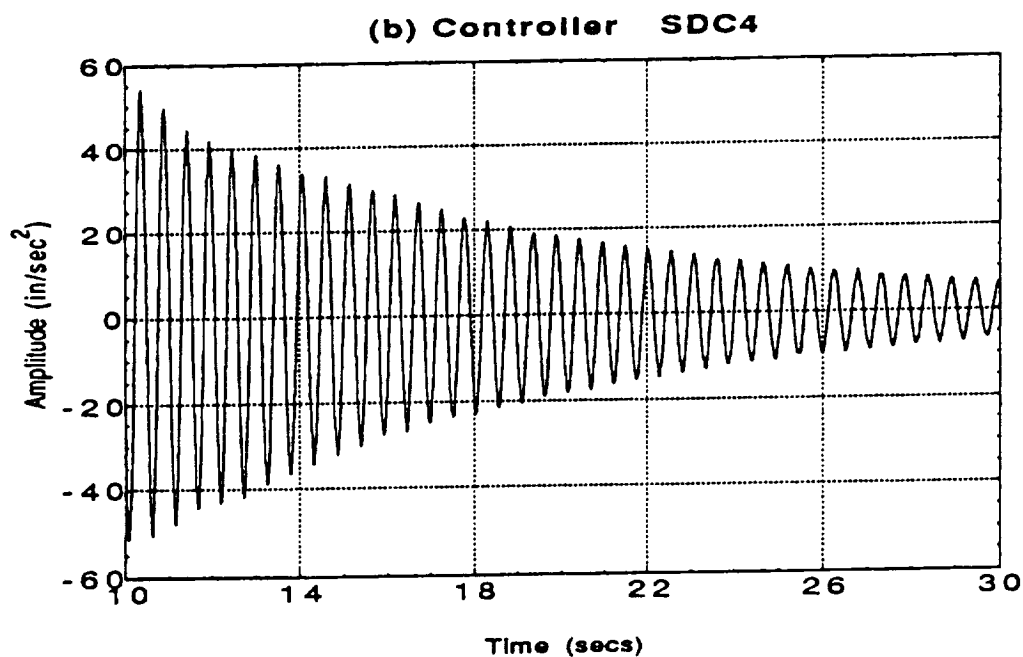
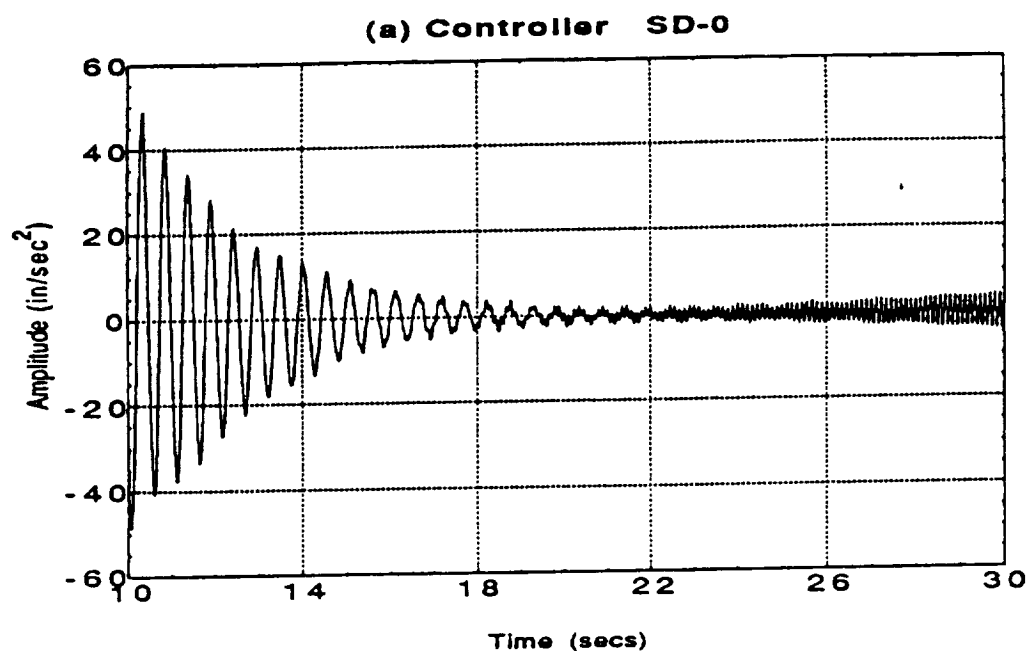


Figure 14: Simulated acceleration response for closed-loop systems with fourth-order Cauer filters (sensor location 1)

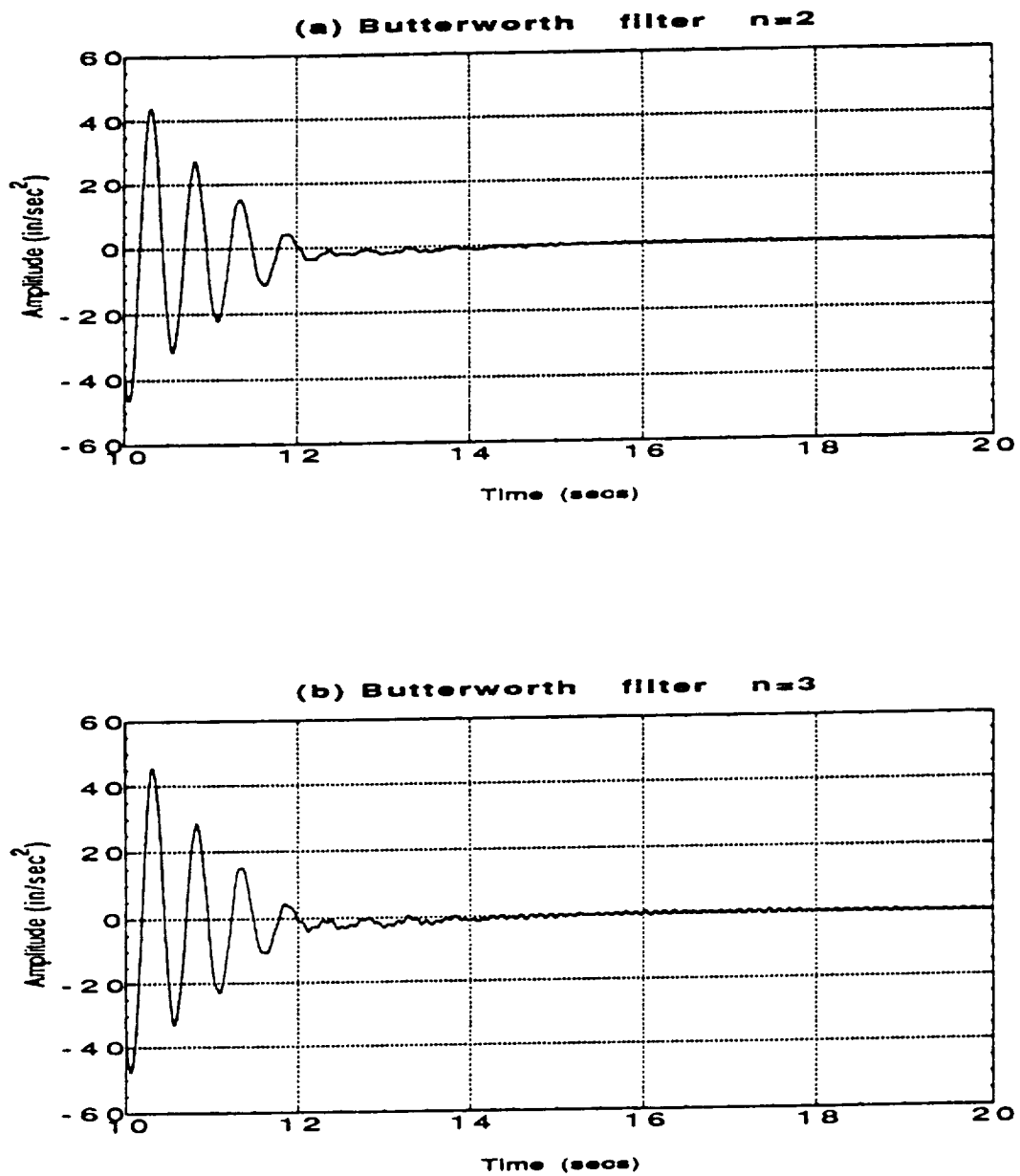


Figure 15: Acceleration response of closed-loop system with controller AVA and Butterworth filters (sensor location 1)

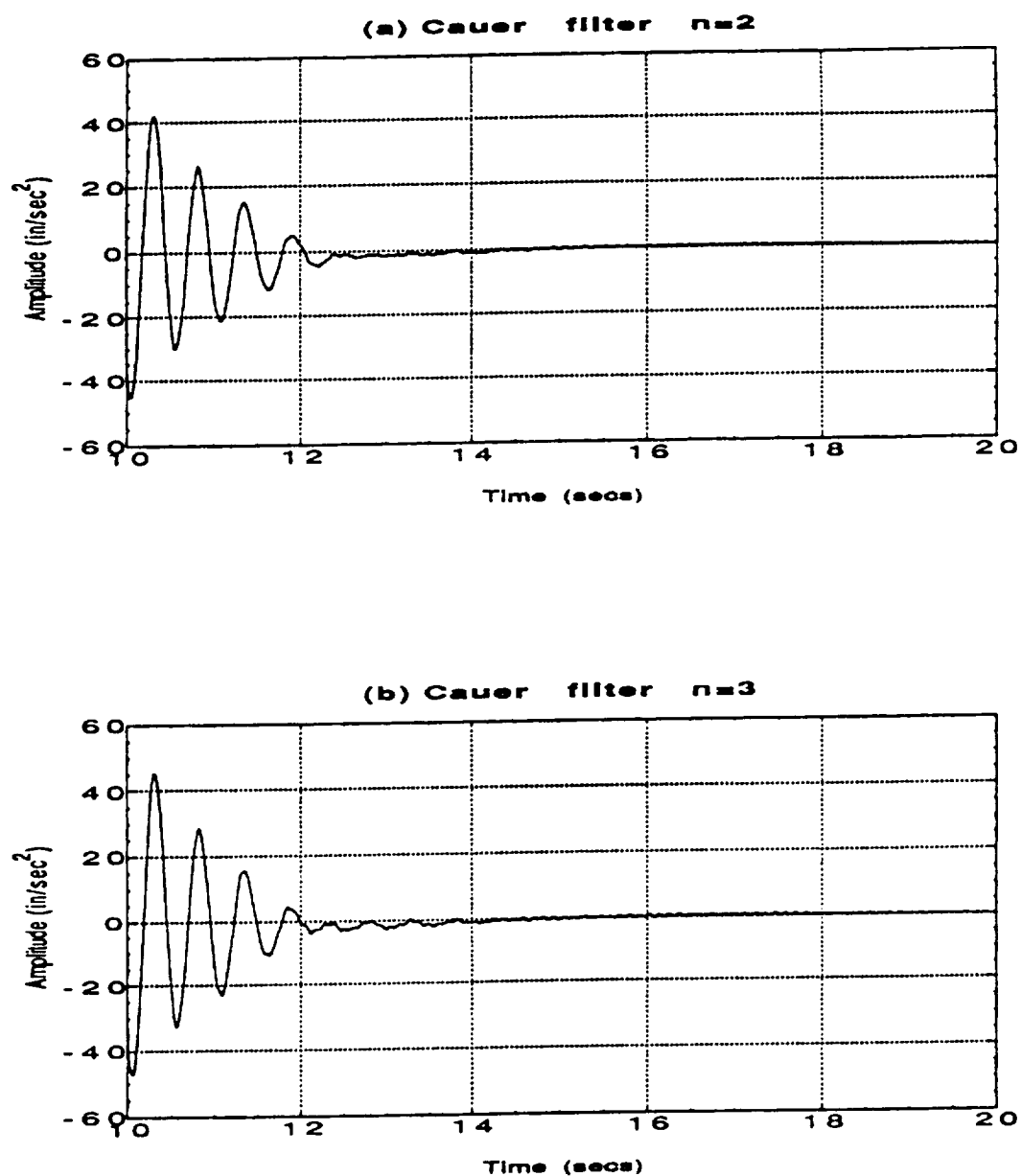


Figure 16: Acceleration response of closed-loop system with controller AVA and Cauer filters (sensor location 1)

VII. Conclusions and Future Activities

Numerical analysis shows that it is possible to use sensor filters in closed-loop control of large flexible space structures (LFSS). The dynamics of higher-order filters must be considered however because the filter phase rolloff causes phase distortion to the output signal. The stability margins of an open-loop system including filters decreases as the order of the filter increases.

Active vibration suppression of a system including the CEM and sensor filters was demonstrated using numerical analyses. Two robust controllers were designed using the static dissipative method and the virtual passive method. Both were developed and evaluated for a system which included filter dynamics. The systems with the static dissipative controllers and second and third-order filters, and the systems with the virtual passive controller and second-order filters remained stable. The systems with higher-order filters became unstable when feedback control was attempted. When the filter dynamics were included in the design model for the static dissipative controller, the corresponding closed-loop systems remained stable. Numerical analysis of closed-loop systems with even sixth-order filters remained stable. However, performance of the controller was greatly decreased.

Laboratory experiments were conducted to validate the numerical analysis. The controllers successfully decreased vibration of the CEM without sensor filters after excitation of several bending and pendulum modes. However, when second-order low-pass filters were implemented digitally to condition sensor output, a 7 Hz mode was excited and became unstable in the closed-loop experiment. The static dissipative controllers which were developed with filter dynamics in the design model were implemented next, and the same 7 Hz mode became unstable. Only the controllers developed with sixth-order filters were implemented successfully. These controllers have very small gains and therefore exerts only small amounts of force on the CEM. Although the higher-order mode was not excited, the controller did not provide much additional damping to the system.

There are several reasons for the difference in results between the laboratory experiments and the numerical simulations, including model uncertainties and discretization errors. Further,

when the low-pass filters are combined with a physical system, the combined system is no longer a pure continuous-time system, thus destroying some of the properties of the original system which were sufficient to guarantee stability of the closed-loop system. A sufficient condition for stability for both the static dissipative method and the virtual passive method is that the open-loop system's mass (M) and stiffness (K) matrices must be positive definite and the damping matrix (D) must be at least positive semi-definite for stability. When the filters are combined with the structure, it may not be possible to express the resulting system as a second-order system with a positive definite M and K , and a D that is at least positive semi-definite. This problem suggests one direction for further research: design a filter which can be combined with a physical structure such that the resulting system can be expressed as a stable second-order system.

The problems with successful implementation of sensor filters for closed-loop experiments in the laboratory suggest several future activities. First, an investigation of how the digital filters are implemented may suggest changes in the real-time computer software. Also, further study of discretization methods for the filter state-space model needs to be made. The results of this study also suggest different approaches for designing a low-pass filter for closed-loop applications.

One approach is to modify the transfer function of a conventional low-pass filter by adding zeros to the numerator in order to compensate for the phase lag caused by the filter. Another approach for low-pass filter design is to formulate the filter as a second-order system. This corresponds to the virtual passive approach for controller design. The filter would be designed as a virtual mass, spring and dashpot system. The mass, stiffness, and damping coefficients would be chosen so that when it is 'attached' to the physical structure, the mass and stiffness matrices of the combined system would be positive definite, and the damping matrices would be at least positive semi-definite.

Control law design for large flexible spacecraft is becoming more challenging as these structures increase in complexity. Typically such systems contain many flexible modes densely packed within a small frequency range. Also there are many unmodeled modes which may affect stability of the closed-loop system. Closed-loop control requires output from sensors for

information about the modes to be controlled. Sensor filters which attenuate unwanted frequency components would greatly facilitate this process. The successful design of sensor filters which do not destabilize closed-loop systems may greatly simplify the control design process.

VIII. References

- [1] Pappa, R. S., Schenk, A., and Noll, C.: "ERA Modal Identification Experiences with Mini-Mast", presented at the 2nd USAF/NASA Workshop on System Identification and Health Monitoring of Precision Space Structures, Pasadena CA, March 1990.
- [2] Hallauer, W. L., Skidmore, G. R., and Gehling, R. N.: "Modal-Space Active Damping of a Plane Grid: Experiment and Theory", *Journal of Guidance and Control*, Vol. 8, No. 3, Aug. 1984.
- [3] Kumar, R. R., Cooper, P. A., Lim, T. W.: "Sensitivity of Space Station Alpha Joint Robust Controller to Structural Modal Parameter Variations", presented at the AIAA Guidance, Navigation and Control Conference, New Orleans, LA, August 1991.
- [4] "Controls Structure Interaction Program Plan, Review Copy", prepared by the Controls Structure Interaction Program Office, Langley Research Center, Hampton, VA, May 18, 1988.
- [5] Belvin, W. K., Elliot, K. E., Bruner, A., Sulla, J., Bailey, J.: "The LaRC CSI Phase-0 Evolutionary Model Testbed: Design and Experimental Results", *Proceedings of the Fourth NASA/DOD Controls/Structures Interaction Technology Conference*, January 1991.
- [6] Meirovitch, L.: *Computational Methods in Structural Dynamics*, Sitjhoff & Noordhoff International Publishers, Rockville, MD, 1980.
- [7] Nimmo, N., "Sensor Filter Designs for Large Flexible Spacecraft", M.S. Thesis, North Carolina State University, 1991.
- [8] Joshi, S. M., Maghami, P. G., and Kelkar, A. G.: "Dynamic Dissipative Compensator Design for Large Space Structures", presented at the AIAA Guidance, Navigation, and Control Conference, New Orleans, LA, Aug. 1991.
- [9] Juang, J. J. and Phan, M.: "Robust Controller Designs for Second-Order Dynamic systems: A Virtual Passive Approach", NASA TM-102666, Langley Research Center, Hampton, VA, May 1990.
- [10] Bruner, A. M., Belvin, W. K., Horta, L. G., and Juang, J. J.: "Active Vibration Absorber for the CSI Evolutionary Model: Design and Experimental Results", presented at the AIAA 32nd Structures, Structural Dynamics & Materials Conference, Baltimore, MD, April 1991.

Appendix C
Start-Up Procedures and Hardware/Software Notes

I: Operational (Start-Up) Procedures.

This system assumes the user has an MS-DOS environment using an IBM-compatible AT 286-based computer (upgrades of 386 or 486-based machines would require minor revision of the software). Further the software requires the Franklin C-51 compiler (see software license agreement section below) and the Dallas software support for the DS2250 (see license agreement section below). Both of these packages are on the floppy disc provided to NASA Langley Spacecraft Dynamics Branch.


Upon entering the DOS environment, simply load file "tf.Bat" and execute. This file prompts the user for transfer function coefficients (stored in an include file). Next the control program is compiled and downloaded to the microcontroller. The load controller (DS2250) takes over and immediately execute the control program.

II. Software Limitations

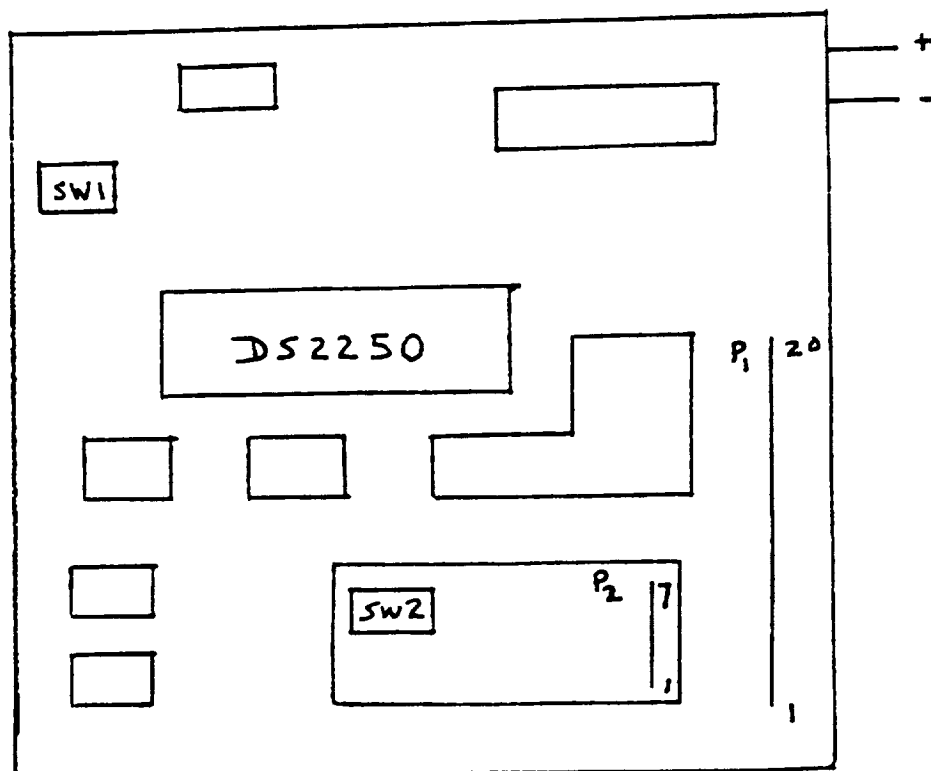
Continuous-time transfer functions are limited to fifth order (due to the direct look-up table for the bilinear-z transformation) and discrete-time transfer functions are limited to 15th-order. To change the discrete-time transfer function order, simply modify the DEFINE statements located at the end of the tf.BAT file to include more coefficient terms (see Appendix A). To change the order of the transfer function in the continuous-time case is more involved. It may be easier to perform the discretization operation off-line (using MATLAB, for example), and enter the controller/filter in discrete-time form.

III. Hardware Issues

The bandwidth of the DPU is subject to the data acquisition system and control algorithm being tested. The data acquisition system has a limiting constraint at the A/D end of the system of 10KHz (the D/A is much faster). The DS2250 has a 12MHz clock. A +5v supply and a $\pm 12v$ voltage reference power supply are required to power the DPU.



A switch is located on the DPU to transfer control from the supervisor system to the load controller. Be sure that this switch is turned OFF when downloading and/or when SW1 is in the stop position (see Figure 11).



P1

20: analog in 0→5v

17: analog out - 10 → + 10v

7-8: GND

4-5: + 12v

1-2: - 12v

SW2: output switch

off ↑ ↓ on

SW1: program switch

stop ↑ ↓ run

Figure 11: Circuit board lay-out

IV. Site License Agreements

In order to fully employ the DPU software, site license agreements must be purchased.

Please contact:

- (a) Franklin Software Customer Service
888 Saratoga Avenue, Suite #2
San Jose, CA 95129

Phone (408) 296-8051

for the C-51 C-compiler

- (b) Dallas Semi Conductor
4401 South Beltwood Parkway
Dallas, Texas 75244-3292

Phone (214) 450-0448

for the DS2250 software or upgrades if desired.